ACCURACY OF A LEARNER’S EDITS, TO THEIR LEARNER MODEL

by

Mark Britland

A thesis submitted to

The University of Birmingham

For the degree of

Mres Electronic, Electrical and Computer Engineering

Abstract

The thesis investigates the accuracy of which a learner can edit their learner model. The research considers how ability of a domain, self assessment skills and learning material could help the learner in editing their learner model. To help identify the accuracy of a learner's editing, erroneous data has been placed into the learner model, to see whether learners can edit their learner models and demonstrate an understanding of their learning. Erroneous data is not normally inserted into a learner model, and in this case is done automatically through the learner modelling process, with the participants having no prior knowledge of this. The learner model developed here is a model of strength of beliefs, where the learner gets to edit their strength of belief. The results showed some interesting findings, with the most noticeable that the majority of participants were able to successfully edit any erroneous misconceptions, whilst none could correctly edit any erroneous knowledge statements inserted.

Acknowledgments

I would like to thank my supervisor Dr Susan Bull for all of her help and guidance, without her I would not have managed to do this

Table of Contents

Table of Figures

Table of Tables

Chapter 1 – Background Research

This chapter introduces the background of intelligent tutoring systems, learner modelling and open learner modelling. These areas are essential for the development and understanding of the research. Along with these background areas it is important to also consider alternative factors which will influence the participants in the research: consider the learners ability (strong or weak) and self assessment skills.

1.1 Intelligent tutoring system and learner modelling

An intelligent tutoring system (ITS from this point forward) is a computer based, adaptive learning environment which usually consists of a domain model (model of the expert understanding – C programming in this research), learner model (student model – this is a model of the learner's knowledge) and teaching strategies (exercises, guidance, feedback from the system, according to the learner model) [5 & 26]. The modelling process is typically achieved by a learner answering a series of questions or problem solving on a particular domain; the system can then infer a model based on these answers, and give the appropriate feedback or guidance for the individual learner [27].

Both the domain model and learner model are usually kept hidden in an ITS and are used to generate feedback for the learner, which can be used to help the learner further themselves, such as: generating exercises for the learner; or suggested learning material. ITSs, through this modelling process can offer an individual learning experience which can be described as trying to care [26], through a learning by doing interaction [28 & 32]. This experience is only possible through the system's understanding of both the domain model and the learner model, giving the learner an individual experience. It is this understanding that gives an ITS adaptive features, offering the learner a tailor made learning experience, designed specifically to improve their learning.

Learner modelling is the core [32] of the ITS and is what allows ITSs to achieve an individual experience based on a learner's knowledge and skill [3]. There are various modelling techniques available to the system implementer. For the purpose of this research we use a perturbation modelling technique. This technique allows the system to model misconceptions of the user whilst also and importantly modelling their knowledge which, in turn, should lead to better feedback (in an ITS).

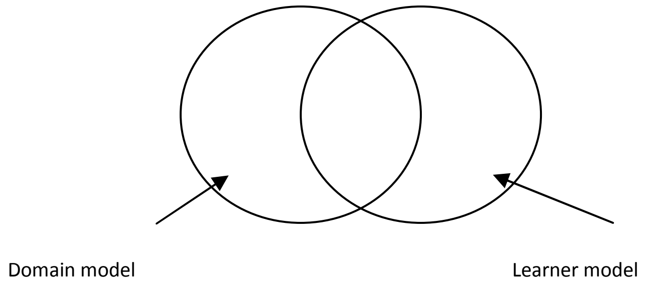


Figure 1 Perturbation modelling [2]

Figure 1 shows graphically the concept of the perturbation modelling technique. Here the more knowledge the learner gains, the more overlap there is with the domain model. The reason this type of technique has been used is because it allows for the inclusion of misconceptions, those parts of the learner model which do not overlap the domain model. Another modelling technique is the overlay technique, here the expert domain is first modelled as a correct set of skills. The student model can then be built based as a sub set of correct and incorrect rules on the domain (expert) model (15). The problem with this is that misconceptions cannot be modelled and as such this technique could not be considered as misconceptions need to be modelled for the research.

The modelling process can only begin once an interaction has taken place with a system and its interface; as stated above this is normally achieved through a series of questions about a particular domain, or attempts at problem solving or other exercises. The domain is often one with a formal structure, one with an underlying set of rules and principles for example C programming. This domain also has a range of common misconceptions which can be stored in a misconceptions library [5 & 27].

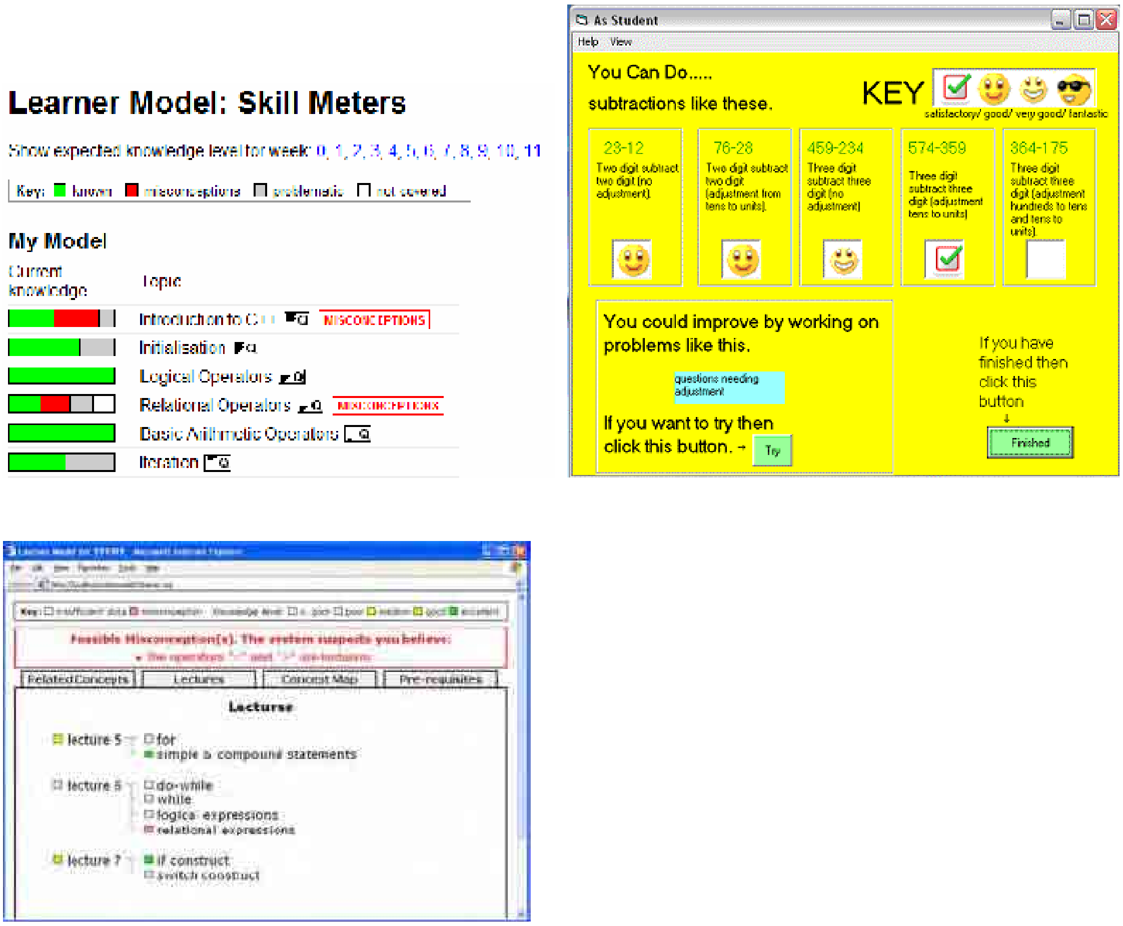
There are however traditional problems associated with ITSs: adjusting the environment to meet the needs of the learner(s) and determining what to say to learners and when to say it [3]. Even though they consider both the learner model and domain model with the aim of individualising the interaction, it is not always possible to understand all the needs of all learners. While traditionally learner models have been kept hidden from the learner and used for system adaptation, there is now a field of research known as open learner modelling (OLM from this point forward), where the learner gets to view their learner model, which typically includes their knowledge and can also include misconceptions and other difficulties. A main aim of the OLM is to help the learner better understand themselves and to promote reflection [6].

1.2 Open learner modelling

Open learner modelling potentially has many benefits for the learner, including: the ability to reflect on one's own knowledge, misconceptions and difficulties which in turn could lead to an enhanced learning experience [16 & 24]. This self reflection process can lead to the learner gaining a better understanding of themselves.

OLM is a relatively new concept in educational technology. Through opening the learner model, it is hoped that it will raise the learner's awareness of their knowledge, difficulties and misconceptions (properties of a learner model), leading to enhanced learning [4]. Opening the learner model offers a unique relationship with the target domain which is otherwise unavailable, possibly encouraging the learner to reflect on their learning, more so than they usually would [16]. Through opening the learner model, Kay (16) identifies four questions learners may seek to answer when viewing their model: What do I know?, How well do I know topic 'X'?, What do I want to know?, How can I best learn 'X'? These are all questions which would point to an enriched understanding of one's own learning.

OLMs can be developed in two ways, as mentioned previously they can be part of a system along with an ITS, but can also be independent (IOLM). Here, the learner just gets an OLM as feedback (based on interaction with the system) and no teaching strategies, like they would get if it was part of an ITS. This research will be using an IOLM, as it will be used to look at how learners interact with their learner models and the accuracy of those interactions.



Learner Model: Skill Meters

Show expected knowledge level for week: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

Key: 🟩 known 🟥 misconceptions ▨ problematic □ not covered

My Model

Current knowledge	Topic
🟩🟥▨	Introduction to C++ MISCONCEPTIONS
🟩▨	Initialisation
🟩	Logical Operators
🟩🟥▨□	Relational Operators MISCONCEPTIONS
🟩	Basic Arithmetic Operators
🟩▨	Iteration

As Student

Help View

You Can Do.....

subtractions like these.

KEY ☑ satisfactory/ good/ very good/ fantastic

23-12	76-28	459-234	574-359	364-175
Two digit subtract two digit (no adjustment).	Two digit subtract two digit (adjustment from tens to units).	Three digit subtract three digit (no adjustment)	Three digit subtract three digit (adjustment tens to units)	Three digit subtract three digit (adjustment hundreds to tens and tens to units).
☺	☺	☺	☑	□

You could improve by working on problems like this.

questions needing adjustment

If you want to try then click this button. → Try

If you have finished then click this button ↓ Finished

Key: □ insufficient data □ misconception   Knowledge level: □ v. poor □ poor □ medium □ good □ excellent

Possible Misconception(s). The system suspects you believe:

- [illegible]

Lectures

- lecture 5
  - □ for
  - □ simple & compound statements
- lecture 6
  - □ do-while
  - □ while
  - □ logical expressions
  - □ relational expressions
- lecture 7
  - □ if construct
  - □ switch construct

Figure 2 – Example learner models

Figure 2, shows several example learner models from various systems [4, 5 & 23], these figures demonstrate a few different implementations of learner models. All three of the LMs show the learners how they are currently doing with a specific domain, topic or concept and as can be seen, shown in many different ways. This opening of the LM is why the learner gets that individual learning experience that would not otherwise be possible, necessarily in another learning environment.

Different types of OLM can be implemented and some have the benefit of the learner interacting with their learner model to influence it, which may lead to a more accurate model [9 & 26]. We here introduce five of the main types of open learner model as defined by Bull and Kay [12]:

- The inspectable learner model (full system control). The inspectable learner model is purely for the learner to look at their learner model, and use it to help identify what knowledge they have and possibly gaps (in knowledge) and misconceptions. [5, 9 and 23]

- There is the co – operative model (the modelling is shared by both the learner and system) so here the modelling process is shared, some things are modelled by the learner and some by the system [31]

- The negotiated learner model is where the modelling process is jointly negotiated by both system and user. However, a negotiation process can take place if either party disagrees with what is being represented in the learner model. [2, 20 & 21]

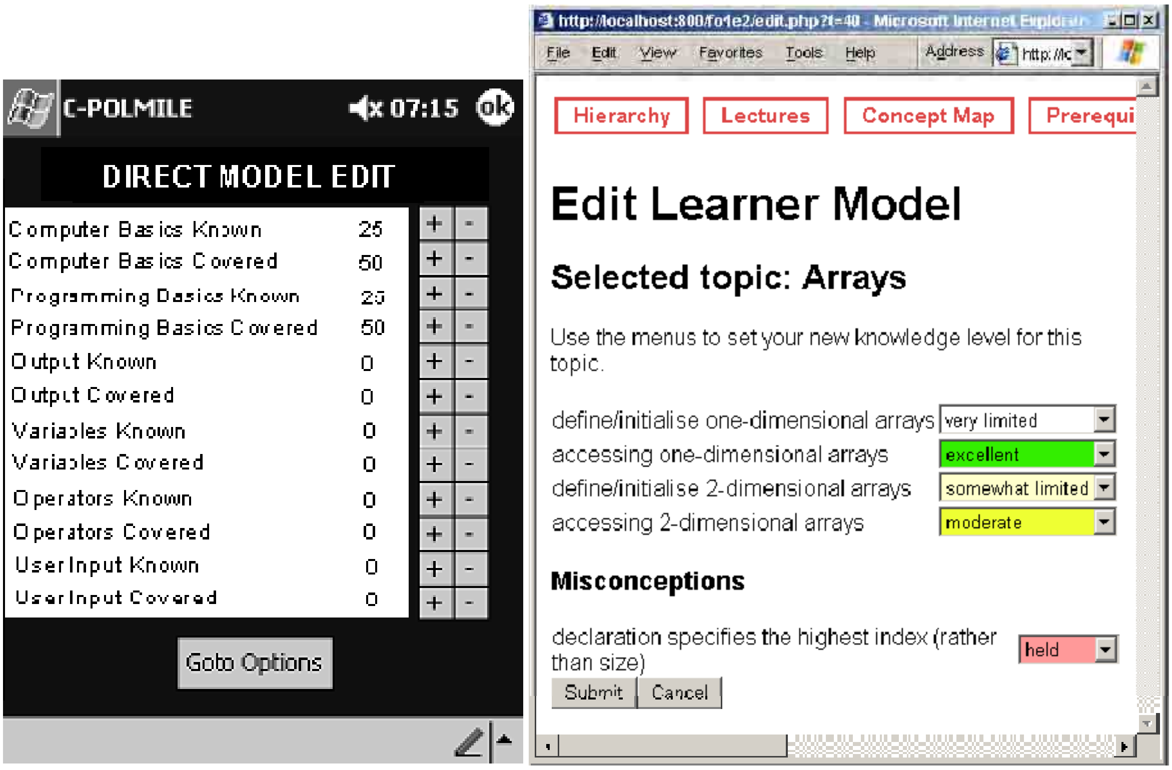
- The persuasion model is where it is down to the learner to try and persuade the system that they have a different level of understanding to the one being shown by the system. This is normally achieved by answering a series of questions on a specific topic where the learner believes they have a different level of understanding to what the system is showing, to demonstrate to the system that their own self assessment is more accurate. If the learner is then able to successfully persuade, the model will be changed to what the learner believes [5].

- Finally there is the editable learner model. The learner will have a model presented to them by the system based on their ability to answer questions or complete other problem solving tasks, on a particular domain, as in other approaches. The editable learner model then gives the learner total control over their learner model and allows the learner to change it as they wish. This is the type of learner model this research focuses on [2 & 23].

Opening the learner model to the learner gives the learner more control over their learning and it can be argued that it improves learning effectiveness [26]. However as mentioned in the editable learner model, the learner here not only has control over the learning process, but also control over the LM contents. Allowing the learner total control may improve the satisfaction they gain from using the system. This point is reiterated by Kay [21], as it provides the opportunity to take responsibility for the content of the model, which could lead to a more accurate learner model. How accurately the learners interact with their LM will be the strong focus of the research.

1.3 Editable learner model

The editable learner model is an OLM of which the student has final say over the contents of the learner (student) model. The modelling process is performed by the system and like other OLM systems a model is presented to the learner. However, using an editable learner model allows the student to edit their learner model to represent what they believe reflects their knowledge or learning (total learner control).



DIRECT MODEL EDIT

Computer Basics Known	25
Computer Basics Covered	50
Programming Basics Known	25
Programming Basics Covered	50
Output Known	0
Output Covered	0
Variables Known	0
Variables Covered	0
Operators Known	0
Operators Covered	0
User Input Known	0
User Input Covered	0

Edit Learner Model

Selected topic: Arrays

Use the menus to set your new knowledge level for this topic.

define/initialise one-dimensional arrays	very limited
accessing one-dimensional arrays	excellent
define/initialise 2-dimensional arrays	somewhat limited
accessing 2-dimensional arrays	moderate

Misconceptions

declaration specifies the highest index (rather than size) held

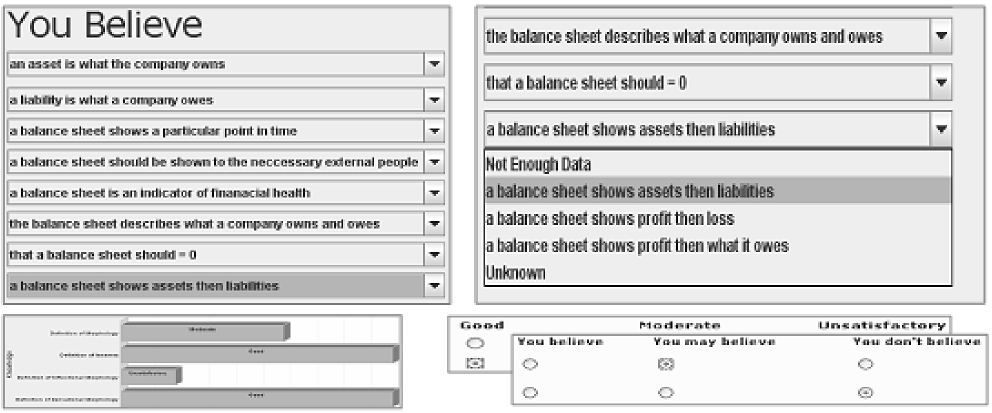
Figure 3 – Editable learner model examples from C-Polmile [8] and Flexi – OLM[23].

Figure 3 shows two examples of editable learner models and their interfaces. The two editable models allow the learner to edit in different ways; there is no one best way to allow the learner to edit. The example on the left allows the learner to edit by pressing the + or – button and the figure on the right allows the learner to edit by selecting the drop down box and changing its contents based on what they believe their understanding of a particular concept within a topic is. Importantly both allow the learner to edit the model and take complete control over their learning.

Previous work [23] has identified some fundamental reasons why the learner may wish to edit their learner model. On initially accessing the system, the learner wishes to edit their learner model, thus avoiding questions about topics they believe they already know. The learner suddenly grasps a concept and wants their model to reflect this without having to answer further questions. Finally, the learner believes they have guessed a series of questions, and wants to edit the model as it shows a higher understanding than they believe they have. All of these can be seen as advantages as they allow for instant updates (accuracy), and they give the learner total control. This research aims to find out whether learners are able to edit their learner models accurately.

The main purpose of being allowed to edit the learner model is to help achieve a more accurate learner model [9]. This will be demonstrated as the student will have total control over their learner model, and they can edit it however they see appropriate. Whereas in previous mentioned OLMs the system has final control, here the learner has control and can adapt the model to what they believe their learning demonstrates (not necessarily what the OLM is showing them).

This total control in terms of the research is good for analysing the ability of the students to accurately edit their learner models (in a way they believe best reflects their learning), as it will not affect them once their participation has finished with the OLM as it is just one research session for a subject they have previously studied. If however this OLM was part of an ITS offering feedback (teaching strategies) changing the LM could affect this. The feedback then given could be detrimental to the learning of the student, because if they have edited their learner model to not accurately reflect their level of understanding the feedback could be too complex, or material the learner already understands. This accuracy will be tested in the research as erroneous data will be inserted into the learner model, whether it is misconceptions or knowledge.



You Believe

an asset is what the company owns

a liability is what a company owes

a balance sheet shows a particular point in time

a balance sheet should be shown to the neccessary external people

a balance sheet is an indicator of finanacial health

the balance sheet describes what a company owns and owes

that a balance sheet should = 0

a balance sheet shows assets then liabilities

the balance sheet describes what a company owns and owes

that a balance sheet should = 0

a balance sheet shows assets then liabilities

Not Enough Data
a balance sheet shows assets then liabilities
a balance sheet shows profit then loss
a balance sheet shows profit then what it owes
Unknown

Figure 4 – editable learner models with erroneous data.

Figure 4, the editable learner models developed in [9] are all OLM examples where erroneous data has been inserted into the LM. The erroneous data in these systems is also unknown to the learners, and the different OLMs were implemented for different reasons. Telling the learners there was erroneous data in their LM would stop them from interacting with the LM truthfully and possibly cause them to try and edit what they believe is the erroneous data. The learners not knowing about the erroneous data in this research will highlight the participant's accuracy of their edits, as they should be truthful edits based on their learning, not trying to edit out erroneous data.

Inserting the erroneous data (not normally done in learner modelling), will aid in determining the extent to which the learner will edit their learner model. This is because having the erroneous data will be a good measure of the ability to edit the learner model, as the researcher knows where the errors are inserted. The user will not know that there is erroneous data in their learner model; therefore how they interact with the learner model will be useful research into the use of editable learner models. If learners do edit the learner model, and they manage to edit erroneous data successfully, it will show that learners do understand themselves as learners and also that editable learner models are a useable tool, providing they do not edit their learner model in such a way that they introduce new errors.

Interesting results should arise from editable learner models, as in the past the concept of editable learner models has shown some interesting outcomes. While in past research [23] participants have expressed users do not feel comfortable having total control over the system. Some of the other participants indicated that they would like the option to edit the learner model [23]. Interesting, is that some students in the past [23] have thought of the OLM as an assessment and therefore considered editing as cheating. It will be interesting to see the interactions and choices made by the participants in this research and see how accurate any edits made are.

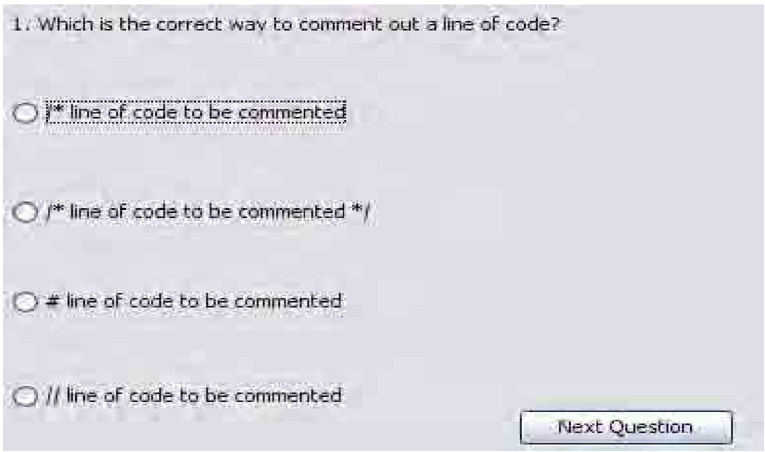
Research into the accuracy of which the learner can edit their learner model includes many issues: frequency of edits; accuracy of edits; extent of edits; whether self assessment, previous OLM experience or participants' ability affects how the leaner may edit their leaner model. To answer these questions an editable learner model was designed and this is described in the following section.

Chapter 2 – System Design

The system will be broken down into separate sections and discussed throughout this chapter. The chapter will focus on the purpose of the different sections and how they were integrated to make a fully functioning system for the purpose of determining the accuracy of the participants' edits.

2.1 Questions

To be able to interact with their learner model, the learner must first answer questions on the different areas within the domain, so the system can model the learner. There are 6 areas, each with 10 questions (Full set of questions in Appendix). This limited number is for the purpose of the research, if it were to be deployed for educational purposes it would need more questions. The learner has a total of 4 answers to choose from and can select 1 answer per question as shown in Figure 5.



1. Which is the correct way to comment out a line of code?

○ /* line of code to be commented

○ /* line of code to be commented */

○ # line of code to be commented

○ // line of code to be commented

Figure 5 - question screen showing four answers and next question button.

As can be seen in figure 5 multiple choice questioning was chosen for the initial investigation, the participants were able to select one answer per question, which they feel was most accurate based on the teachings of the C programming module (explained later). Multiple choice questions make it easier and quicker for the learner to interact with the system and understand what is expected of them when answering the questions. Finally any known misconceptions can be included in the responses.

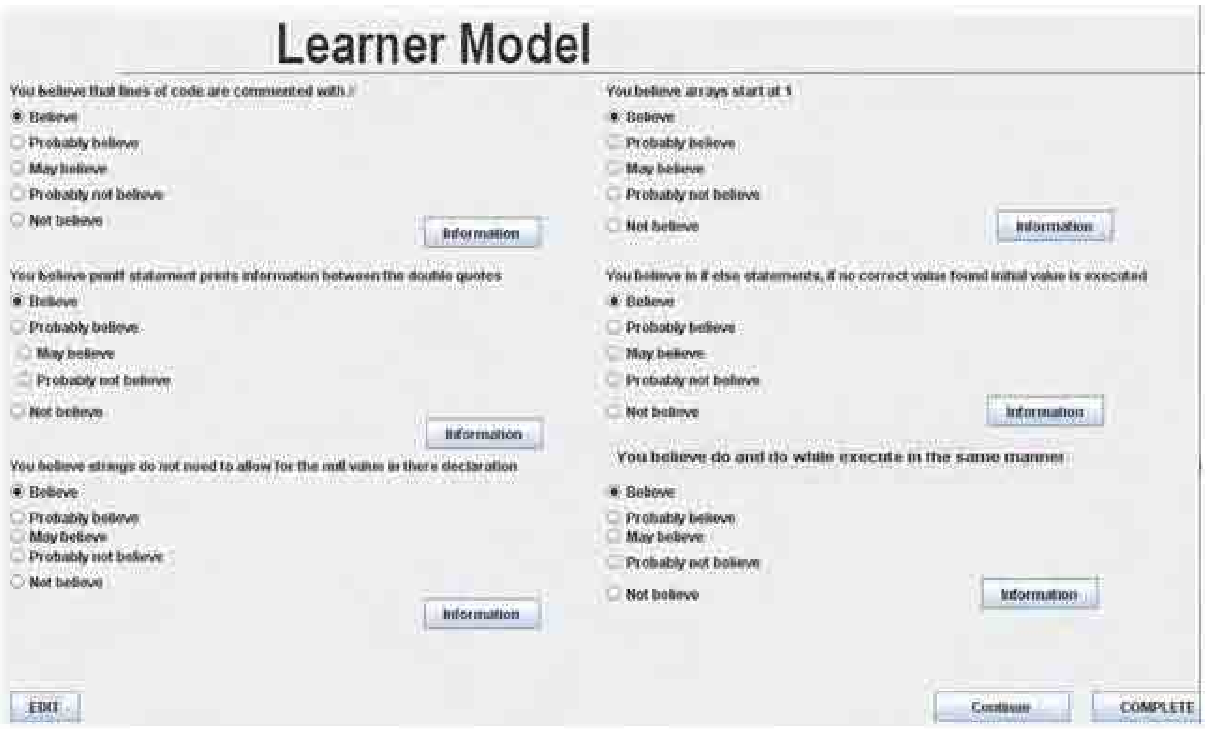
The learner is free to view the model at any time during the questioning process, except during the first 5 questions of a concept as there is insufficient data to model. They may view the information pages before answering questions, or they may answer questions without first viewing the information, to offer flexibility in the learning approach.

2.2 Learner Model

The learner model is produced as mentioned earlier, through the learner answering multiple choice questions on topics within the domain, these answers are then used to model beliefs, and shown to the learner in their LM. This learner modelling process is carried out by modelling, based on the five most recent answers to questions the learner has given within a topic of the domain (e.g. 1-5, 2-6 or 4-8 a series of questions in a topic), which is also used in OLMLETS and Flexi –OLM [6 & 24]. This gives the system the most up to date representation of the learner's learning, which can then be shown to the learner in the OLM. This is achieved in the system by developing an array based on the answers given to a series questions within a concept, the array is used to store a value of the answers that the learner has input. The LM is then inferred from the answers given by the learner, and the appropriate statement from the underlying LM will be shown to the learner.

The type of learner model that is developed for the research is a model of strength of beliefs, based on a Likert scale (22) design. There is a 5 point strength of belief scale available for the learner to edit. Above the scale there is a statement, in the form of "You believe 'x'", this statement is a reflection of a particular concept within the domain (misconception, difficulty or knowledge statement).

The initial belief shown on the scale will show the system's interpretation of what it believes the learner understands. The statement and strength of that statement are based on the learner's answers to questions on a topic within the domain (system interaction); the five points of the scale are: believe, probably believe, may believe, probably not believe and not believe. The scale will be the editable part of the learner model. A scale will be used for the learner model in this research because; most people will have had some previous experience interacting with a Likert type scale, easing interaction through familiarity.


Learner Model

You believe that lines of code are commented with //

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe printf statement prints information between the double quotes

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe strings do not need to allow for the null value in there declaration

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe arrays start at 1

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe in if else statements, if no correct value found initial value is executed

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe do and do while execute in the same manner

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

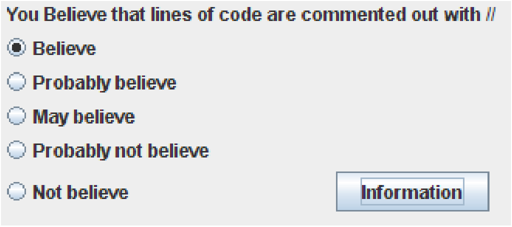
Figure 6 – The Learner Model

Figure 6 shows a learner's LM, this is what the LM would look like to the learner, once the learner has answered enough questions about each concept for the system to build a model for this aspect their knowledge. In order to achieve these different LMs that will occur as different learners interact with the system, the system has an underlying LM which has different statements that will be displayed on the learner's LM, depending on the answers the learner gives to the questions. This underlying LM within the system which is used to infer the learner's LM based on the learner's responses to questions, is important as it allows for the individualisation of the LM for the individual learner's interacting with the system.

Whilst using this system both a single misconception and single piece of knowledge are inserted into the learner's learner model. Of the six sections of the learner model, two had erroneous data in them. The first piece of erroneous data to be inserted was inserted once the learner had answered sufficient questions in the system for the learner to be modelled. The second piece of erroneous data was inserted into the model once the learner had answered questions in all sections of the learner model. This was done to limit the amount of erroneous data in the learner model and to maintain learner trust in the model.

The system architecture can best be described in three sections; firstly there is the input, the answers that the learner gives to the multiple choice questions they are answering. Then there are the calculations, the main one being an array which is used to store the learner's 5 most recent answers on a topic, which the system then uses to infer the LM statement and level of belief for this statement. Finally the output, which consists of the LM and 'behind the scenes' logs. The logs are made up of all the answers that the learner selected, any edits the learner made and any learning material that the learner chose to view. These logs show how the learner interacted with the system, and will be used in the evaluation of the system.

In total there are six LM statements, and each statement is linked to a particular concept of a topic, within the domain. The reason a belief statement was chosen was because it is considered better for prompting learner reflection and learning as it is probably easier for a learner to understand a statement about what they believe e.g. "you believe X", rather than have a title e.g. "arrays" followed by a strength of belief. This feature aims to prompt more interaction as the learner is able to reflect on the belief statement rather than an area title.



You Believe that lines of code are commented out with //

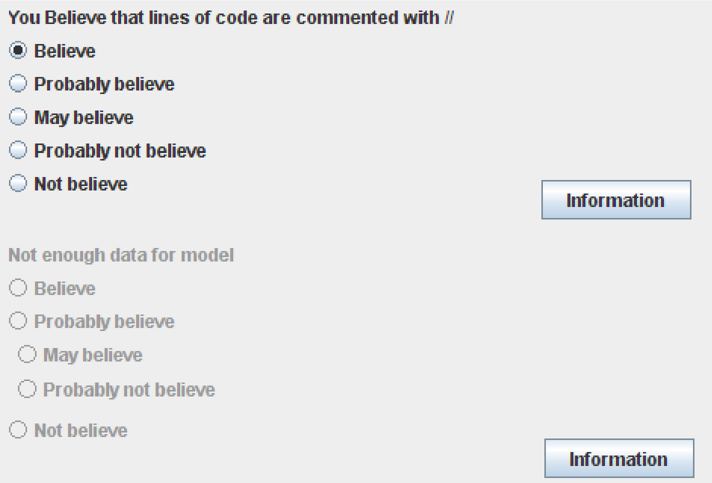
● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

Figure 7 (Part of the OLM), statement at the top, level of belief at bottom, information button bottom right corner.

Figure 7 shows a section of the open learner model, there is the statement at the top and then the strength of that belief at the bottom. The part of the model that is editable is the strength of that belief, so the learner can click on one of the radio buttons to the left hand side of the strength of belief and the model will be edited accordingly.

A five point scale will be implemented in the system as in other work [30]; a 5 point scale has been suggested to be an optimal number of points for a scale, for a person to judge themselves on. The more points on a scale the more complex it becomes, and the fewer points on a scale the decisions made have a greater influence. The scale will enable us to monitor and compare the appropriateness of the edits made by the learners, as it gives a clear indication of how the learner interacts with the model giving a definitive movement on the scale. If for example the way in which the learner could have edited their learner model was more open (not set to 5 points), the interpretation of how appropriate an edit was, would not be as concise.

A key question of this research is the how the learner interacts with their learner model, more importantly the editing of the learner model. As there are 6 topics in the domain, the learner only has the option to edit those topics which they have answered questions on, as in other sections there will be no data modelled and therefore, there would be no way to measure the accuracy of any edits. To make the learner aware of sections they are able to edit, when wishing to edit a part of the model they need to press the edit button. Once this has been done the available sections to edit (ones where the learner has answered questions) become highlighted making it apparent to the learner what they should be looking at. This highlighting process is designed to draw the learner's attention to the parts of the model they can interact with, with the aim of encouraging interaction with it where appropriate as shown in Figure 9 below.



You Believe that lines of code are commented with //

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

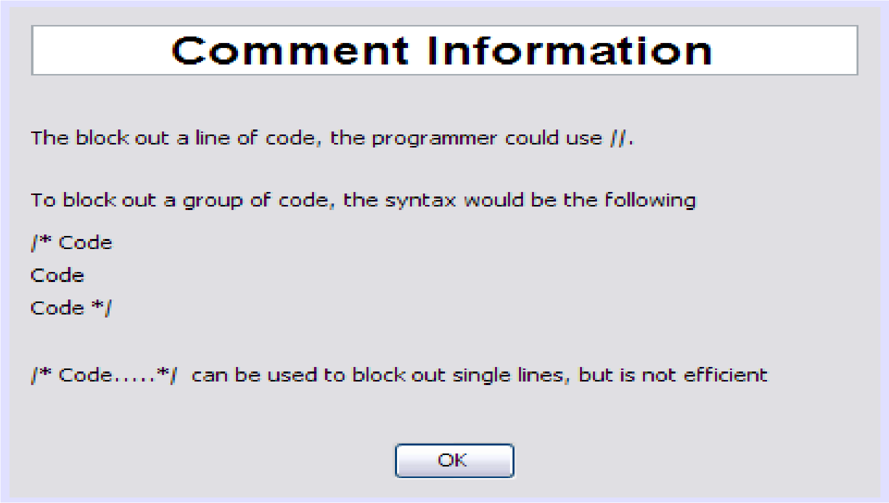
Not enough data for model

○ Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

Figure 8 – highlighting, when the edit button is clicked

2.3 Information screen

On the open learner model Figure 6 it is clear to see six buttons with the word "information" written on them. There are six buttons, because as mentioned there are six concepts which are used from the C programming domain. The information button has relevant information about the area of the domain it is placed next to, as shown in Figure 6 by the commenting information box. If the learner clicks on it a pop up box containing information is presented to the learner.



Comment Information

The block out a line of code, the programmer could use //.

To block out a group of code, the syntax would be the following

/* Code
Code
Code */

/* Code.....*/ can be used to block out single lines, but is not efficient

Figure 9 - Information box, showing information relating to section of the domain model

This information may help the learner in understanding the belief statement in their learner model. The aim of the information boxes is that they should prompt further learner reflection as they will be able to use it as a learning tool along with the OLM. They could use the information buttons for a variety of reasons, validation of thought, try logically working out if the belief statement in the learner model is true or read it to boost their own knowledge.

2.4 System output (Logs)

The final part of the system design is to consider the logging of the information and the erroneous data. Erroneous data is not normally inserted into the learner model, but for the purpose of this research, there has been erroneous data automatically inserted into the learner model, which aims to help establish whether learners really understand their learning and will help to identify the extent to which the learners' edit their learner model. This helps demonstrate the learner's understanding of their learning as the output of the erroneous data will be a contradiction of what the LM statement should be, based on the answers given to the questions in the system.

Then monitoring how the learner interacts with the sections of the LM where the erroneous data has been inserted will help in identifying how well a learner understands their learning and how appropriate any edits they make are. The erroneous data was placed on the extremes of the scales; this then shows the extent and appropriateness of the edits to their LM. In the system logs it will be clear to see how many scale point moves the learner makes where erroneous data has been inserted, with a maximum of four scale point moves for the two pieces of erroneous data, 8 scale point moves in total. This will be used as a measure of the appropriateness of the edits to the LM.

The erroneous data for each participant is in the form of one knowledge statement turned into a misconception and one misconception turned into a knowledge statement. The reason there are not too many errors within the LM as trust issues could develop and stop the learner from using the LM [1]. This way it will help the researcher distinguish any learner patterns between editing misconceptions and knowledge. Finally, the logging of the system use is broken up into three sections: log showing when they view the learner model, which also indicates when the learner edits the learner model; log of the edits made to the learner model whilst interacting with their learner model; log of the information boxes they use.

Chapter 3 – Evaluation

3.1 Aims/ Goals

The main goal of the research is to discover the accuracy of a learner's edits to their learner model. This will include such areas as number of edits, accuracy of edits, extent of accurate edits, why people edited and ability to edit inserted erroneous data. Erroneous data is where the answers given are manipulated to show a misconception as knowledge or vice versa. So for example the student may have shown knowledge in the concept of commenting, but their learner model would show they have a misconception in their learning.

The domain of C programming was chosen to investigate the research question, as it is an area which has an underlying set of principles and rules (and common misconceptions) making it ideal to be implemented for an OLM system. The C programming domain was also chosen as the subject is taken by everyone in the Electronic, Electrical and Computer Engineering School at the University of Birmingham and therefore there is greater potential to support a large number of future users, and a sufficiently large pool of participants for the evaluation. The entire domain of C programming was not used as it would have been far too large a learner model to develop with too many questions for the length of the research. Therefore only 6 concepts were used, one for each section of the learner model. Where in the text it refers to knowledge of C programming, this is the learner's knowledge of C programming based on previously mentioned core module taken by all of the participants.

3.2 Participants

The 20 participants were all University of Birmingham students, who have studied in the UK, they were of different ethnicities but all spoke English as a first language, they were all volunteers. They had all completed a module 'Introduction to C programming', all participants must have passed this module as it is core to the learning of any degree within the Electronic and Electrical and Computer Engineering school in the University of Birmingham, this is a first year module. All participants were fourth year, MSc or MRes students.

3.3 Materials

The materials available to the participants were the instructions and use of a laptop which had the system on it. The system consisted of an information screen, question screens and editable learner model. The information screen was where the participant inserted their participant number, previous OLM experience (yes or no, level of use of OLM was in a questionnaire) and level of C programming ability (very strong – very weak on 5 point scale). The question screens, where the participants select their answers to the different concepts. The editable learner model, this is where the learner is able to view their learning and make choices about their learning. A post system use questionnaire (Appendix), where the learner answered a series of multiple choice questions about the research, which included a couple of open ended questions for more detailed explanations.

3.4 Methods

Participants were given the same set of instructions: the instructions introduced them to the study, answering questions on C programming, and finally introduced them to the system, the question screens and editable learner model via a series of screen shots (paper based system) and what all the interactive parts of the system were (Appendix). All participants had sufficient time to read and ask any questions about the instructions or study (as much time as they wanted).

The participants were made aware that their understanding of C programming was going to be questioned in the investigation. They were also clearly told by the researcher that this information would be displayed in an editable learner model, and given information about what an editable learner model is. All participants then read and signed an ethics/consent form (Appendix) showing that the study was carried out according to British psychological society guidelines for studies involving human participants. (BPS)

The next step involves using the editable learner model system; the participant was given a participant number to remain anonymous once the study was in session. All participants were then given unlimited time to interact with the system. Participants were allowed to ask navigational questions whilst using the system. Participants were then given time explore the system, before commencing with the system use for the research. This allowed the participants time to familiarise themselves with the system and see how it worked.

The participants then used the system, once the participants had answered questions in the system the participant then had the option to view their LMs, where they get the option to use all the interactive parts of the system (learning material and model editing).

The final stage of the participation is to fill in a 20 question questionnaire, which was made up of both closed and open ended questions. The closed questions were all based on a 5 point Likert scale, which went from strongly agree – strongly disagree. These questions were all based on their experience with the editable learner model system. The open ended questions were why / why not questions based on the closed questions. Not all closed questions had a corresponding open ended question. There was also a final box where participants could put any additional comments. The information gained in the questionnaires was used jointly with the systems logs, to identify patterns in the learner's interaction with the system.

3.5 - Results

Table 1 – Relationship between edited erroneous data, previous OLM experience and editing learner model if disagreed with it.

Participant	Misconception Edits	Knowledge edits	Previous OLM experience	Used ELM before	Believe Model is accurate	Edit when Disagreed
1	0	0	3	3	4	2
2	4	0	4	4	4	4
3	0	0	1	1	3	4
4	4	0	3	2	4	4
5	0	0	1	1	2	5
6	3	0	4	3	2	5
7	3	0	1	1	4	4
8	3	0	1	1	4	3
9	3	0	3	2	3	3
10	4	0	1	1	3	4
11	4	0	3	2	5	4
12	4	0	3	1	4	3
13	2	0	1	1	3	2
14	3	0	1	1	3	3
15	3	0	3	2	3	4
16	4	0	1	1	3	4
17	0	0	4	2	4	1
18	0	0	4	3	3	1
19	4	0	4	2	4	4
20	4	0	1	1	3	3
Mean	2.6	0	2.35	1.75	3.4	3.35
Median	3	0	3	1.5	3	4
Range	0-4	0	1-4	1-4	2 – 5	1-5

From the interaction logs, table 1 indicates that 75% (15 out of 20) of participants successfully edited the inserted misconception to some extent on the 5 point scale (mean 2.6, median 3). Of the 15 participants 8 edited the misconception the full extent (the maximum number of scale point moves, 4) indicating they did not believe the misconception statement. None of the 20 participant attempted to edit the inserted knowledge.

Of the 20 participants 5 stated they had high levels of previous OLM experience from modules in previous years of study, with one of these participants also having had a high level of editable learner model (ELM) experience. A mean of 2.35 and median of 3 indicate about half have had previous experience with OLM. The mean for ELM was lower at 1.75 and median of 1.5. The level of experience was decided by the participants.

The final section of comparison in this table is believed accuracy of the learner model, and whether the learner edited their learner model if they disagreed with it. The mean for accuracy of the learner model was 3.4 with a median of 3, suggesting that the average participant was undecided about the accuracy of their learner model, shown in a range of 2 – 5. Finally whether the participant edited when disagreeing with their learner model, showed a mean of 3.35 and median of 4. The median suggests participants would generally edit if they disagreed with their learner model, whereas the mean indicates there is some variety, also shown with a maximum range of 1 – 5.

Table 2 - relationship between self assessment, C programming ability and direction of edit

Participant	Perceived self assessment skills	Perceived C programming ability	Actual C programming ability	Direction of changes
1	2	2	2	Weaker
2	3	2	2	Stronger
3	3	3	2	Stronger
4	4	2	2	Same level
5	1	3	3	Stronger
6	3	4	3	Stronger
7	2	2	2	Stronger
8	5	2	2	Stronger
9	1	3	2	Stronger
10	3	4	3	Stronger
11	4	4	2	Stronger
12	5	2	2	Same level
13	2	3	2	Stronger
14	3	3	2	Stronger
15	2	2	2	Stronger
16	4	4	3	Stronger
17	5	3	1	Weaker
18	1	2	1	Weaker
19	3	4	4	Stronger
20	4	3	1	Stronger
Mean	3	2.85	2.15	
Median	3	3	2	
Range	1-5	2-4	1-4	
Chi - Squared	Chi – squared statistical analysis – perceived C programming ability – Actual C programming ability Chi-squared = 5.991 and test statistic is 10.6107 – therefore a significant difference between perceived and actual ability – (Appendix for calculations)			

From the questionnaires and interaction logs, table 2 indicates that participants were generally not confident in their self assessment (SA) skills; potentially affecting how they edit their learner model, as the mean for SA was 3 with a median of 3 (range of 1-5). 7 of the 20 participants indicated that they had strong SA skills, yet none of the participants successfully edited both the inserted knowledge statement and misconception statement (see table 1).

SA is shown in the table with perceived ability of C programming and actual ability of C programming, actual ability was measured based on the learner‘s learner model before they made any edits. The means were very close, 2.85 and 2.15 respectively and the medians were 3 and 2. This would indicate that SA skills were average based on mean and medians of the participants' data.

9 participants perceived C programming ability and actual C programming were the same. The programming ability was judged on what the participants LM looked like prior to any editing. 11 participants had initially over estimated how good they were at C programming. Of the 9 participants correctly stating their level of C programming 3 had said they were excellent at SA (4, 8 & 12), with an overall mean for the 9 participants of 3 and median of 2.

Finally the two above points were considered against appropriate editing of the inserted misconception and knowledge, and whether the learner also made their learner model reflect that of a weaker or stronger learner. We know from table 1 that 75% of participants successfully edited the inserted misconception and no one edited the inserted knowledge. 75% made themselves represent a stronger learner than they actually were, 15% made no alterations to their learner model and finally 10% edited their learner models inappropriately but the changes made reflected a similar level of understanding.

A Chi-squared significance test was performed on the data for what the perceived ability of the learner was and what their actual ability in C programming was. The significance test was done using a test statistic with two degrees of freedom, which has a critical value of 5.991. The Chi-squared significance value calculated (Appendix) was 10.6107, this indicates that the result was significant. This then indicates that the participants were not able to judge their ability of C programming. This is a clearly observable statistic by looking at the two columns in Table 2.

Table 3 – Sections of the learner model edited

Participant	Section 1	Section 2	Section 3	Section 4	Section 5	Section 6	Total edits
1	0	0	0	0	0	0	0
2	4	4	0	0	0	0	8
3	4	3	4	0	2	0	13
4	4	2	0	0	2	0	8
5	0	2	4	-4	0	0	10
6	4	0	4	3	0	0	11
7	3	2	3	-4	0	0	12
8	3	0	0	0	0	0	3
9	3	0	2	3	1	0	9
10	4	0	3	4	0	0	11
11	4	0	3	4	3	0	14
12	4	0	0	0	2	0	6
13	4	0	0	-2	3	0	9
14	3	1	0	0	0	0	4
15	3	3	0	0	4	0	10
16	4	0	3	4	0	0	11
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	4	0	0	4	0	0	8
20	4	3	2	4	3	0	16
Mean	2.95	1	1.4	0.8	1	0	8.15
Median	4	0	0	0	0	0	9
Range	0-4	0-4	0-4	(-4)- 4	0-4	0	0-16

Table 3 indicates that 17 of the 20 participants edited their learner model, showing which sections (concepts) of the learner model the learner edited. The numbers in those columns indicate how many scale point moves the learner made, with 4 being the maximum. The numbers indicate the correct number of movements in relationship to the belief statement of the LM. Correct here stands for academically correct, not whether the editing of the participant was correct.

The minus numbers are where the learner has edited their learner model and the edit is not correct in terms of the concept, for example the statement is a correct knowledge statement for a concept, then the learner edits their learner model from believe towards the not believe point. Out of all the edits made, only three of them were not correct in academic terms.

This comparison gives a more detailed look at how the participants edited the individual concepts (Section 1 – Section 6). The only section that had considerable editing was the first section with a mean of 2.95 (out of 4) so this would reflect a reversal in the belief statement being shown to the learner in their learner model. The editing patterns show a lot more editing in the first three sections with means and medians of 2.95 and 4 for S1, 1 and 0 for S2 and 1.4 and 0 for S3, compared with 0.8 and 0 S4, 1 and 0 S5 and 0 and 0 S6. This shows participants edited a lot more in the earlier sections domain.

Table 4 - Relationship between viewing the learning material and editing the erroneous data.

Participant	Viewed all learning material	Viewed learning material where errors inserted	Viewed learning material of at least one inserted error	Viewed learning material where edited LM	Only edited LM where learning material viewed	Learning material helpful
1	Yes	Yes	Yes	No	No	1
2	No	No	Yes	Yes	Yes	3
3	No	No	No	Yes	No	5
4	No	No	Yes	Yes	No	4
5	No	No	No	Yes	Yes	4
6	No	No	No	Yes	No	4
7	Yes	Yes	Yes	Yes	No	5
8	No	No	Yes	Yes	Yes	2
9	No	No	Yes	Yes	No	4
10	No	No	No	No	No	1
11	No	No	No	Yes	No	4
12	Yes	Yes	Yes	Yes	No	4
13	No	Yes	Yes	Yes	No	2
14	No	No	No	No	No	1
15	No	No	Yes	Yes	No	5
16	No	No	Yes	Yes	No	4
17	Yes	Yes	Yes	No	No	4
18	No	No	No	No	No	3
19	No	No	Yes	Yes	Yes	3
20	No	No	Yes	Yes	No	4
Mean						3.35
Median						4
Range						1-5

Table 4 covers all aspects of the learning material in comparison to whether the learner edited the inserted misconception and/or knowledge. The first column indicates that 4 out of 20 participants viewed all the available learning material in their learner model, (there was learning material for each of the 6 concepts the participants answered questions on). Only 5 of the 20 participants viewed the learning material where both pieces of erroneous data were inserted, in two of the concepts. However, 13 people viewed the learning material of at least one concept where the erroneous data had been inserted in their learner model. 4 participants only edited their learner models in sections where they had viewed the learning material.

The mean for whether people thought the learning material was useful was 3.35 with a median of 4, suggesting that participants found the material on average not useful, reflective with a range of 1 – 5 (minimum to maximum). As seen in previous tables this data is shown in relationship to whether the participants successfully edited the inserted erroneous data, 75% success with the misconception, with no one correctly editing the inserted knowledge.

Table 5 - Relationship between self assessment, C programming ability and viewing the learning material.

Participant	Perceived self assessment skills	Actual level of C programming	Viewed at least one piece of learning material	Learning material helpful	What learner model should look like	What learner model did look like
1	2	2	No	1	2	2
2	3	2	Yes	3	2	4
3	3	2	Yes	5	2	5
4	4	2	Yes	4	2	3
5	1	3	Yes	4	3	3
6	3	3	Yes	4	3	6
7	2	2	Yes	5	2	3
8	5	2	Yes	2	2	3
9	1	2	Yes	4	2	3
10	3	3	No	1	3	6
11	4	2	Yes	4	2	6
12	5	2	Yes	4	2	2
13	2	2	Yes	2	2	4
14	3	2	No	1	2	3
15	2	2	Yes	5	2	5
16	4	3	Yes	4	3	6
17	5	1	Yes	4	1	1
18	1	1	Yes	3	1	1
19	3	4	Yes	3	4	6
20	4	1	Yes	4	1	5
Mean	3	2.15		3.35	2.15	3.85
Median	3	2		4	2	3.5
Range	1-5	1-4		1-5	1-4	1-6
Chi - squared	Chi – squared statistical analysis – What model does look like – What model should look like Chi-squared = 5.991 and test statistic is 39.375 – therefore a significant difference between what model should and does look like – (Appendix for calculations)					

Table 5 shows how SA, actual ability of C programming along with viewing the learning material and how helpful this was impacted upon the participants' learner models. As seen in previous tables the participants mean average for SA was 3 and the mean average for actual C programming ability is 2.15, this showed that the average participant was weak at C programming and their SA skills were considered mediocre.

The participants on average found the learning material not helpful with a mean of 3.35 and median of 4. The potential usefulness of the learning material to the learner is shown in the columns: what the participants' learner models should have looked like (this is a LM with no erroneous data and prior to any edits), and what they did look like (this is a LM taking into account all edits made by the learner). On average the learner models should have represented a weaker learner based on what the LMs looked like before the participants edited. Of the six concepts the average learner demonstrated knowledge in 2.15 topics. Once all the editing had been taken into consideration, the average learner reflected a strong learner with the mean average of 3.85, implying that the learner model now demonstrated knowledge in nearly 4 of the 6 concepts.

A Chi-squared significance test was performed on the data for what the learner model should look like (actual) and what the learner model did look like (expected). The significance test was done using a test statistic with two degrees of freedom, which has a critical value of 5.991. The Chi-squared significance value calculated (Appendix) was 39.375, this indicates that the result was significant. This indicates there was a significance difference between what the participants' learner models should look like and what they did look like. By viewing the results in Table 5 it is clear to see there is a very positive difference in how the participants edited their learner models. All of them made their learner models reflect that of a stronger learner or actually the same as they were, none of the learner models reflected a weaker learner model.

Considering a cross analysis of the tables and individual participant results, there are some further points that can be made about the findings. Firstly considering previous OLM and ELM, the three participants (1, 17 & 18) that did not edit their learner model at all, all had previous OLM experience with a mean for these 3 participants of 3.6. Also all three of these participants had interacted with an ELM but with less experience, a mean of 2.6, suggesting a limited use. Interesting here that two of the participants (17 & 18), viewed the learning material when interacting with their learner model. The three participants (1, 17 & 18), had previous OLM experience and did not edit their learner model. There were two more participants (3 & 5) that did not edit the misconception in their learner model.

3.6 Qualitative results

The open ended sections of the questionnaire allowed for the opinions and thoughts of the participants about their learning experience whilst using the system. In the additional comments section, 3 participants mentioned their C programming skills were not as good as they first believed them to be when stating their level of skill at the beginning of the interaction process “I think I was let down by overestimating what I thought I knew” and programming skills not as strong as I thought – in hindsight should have stated a lesser level of understanding of C programming”.

The next point that was mentioned a lot was the usefulness of the learning material. 17 of the participants had used the learning material, and the reasons for its interaction were validating their learning and using it to edit the model. This was stated by 10 of the participants in the open ended part of the questionnaire, some of the comments were as follows “After viewing material it let me edit my model to show my new understanding”, “It allowed me to see what I did not know” and “Once viewed the learning material I could change it”. Most of the participants stated that they generally found the learning material helpful.

17 of the participants had edited their learner model, as such there were comments stating the usefulness of the editable learner model, some of which were “I could edit to show my new understanding”, “I liked being able to change my responses” and “It gave me the opportunity to change it so I could focus on areas where I would need more help”.

Participants also stated that they would use an OLM again, this was mainly because they found the experience useful and could see the benefits of being shown their learning, “It showed me what I knew and what I didn’t” and “Yes because I haven’t done C programming in years”.

When asked about using the editable model again, although participants had said they found the editing useful (see above), 4 participants made comments reflecting a negative stance towards the fact they could change their learner model with no challenge to this   “No, if kept changing it I wouldn’t know what I’d already understood or learnt”.

One participant was particularly against the learner model experience, and the fact there was a system modelling them and showing an output of their learning, and this participant had no previous experience with OLMs. They stated “I prefer to learn in my own style” and this participant is the same participant who did not like the idea of being able to keep changing the model and losing track of their learning. Finally they stated “I prefer getting a result like a grade or test result”. Another participant had said “I am not familiar with OLM”, as to why they did not like the experience.

There were three participants all with previous OLM experience who chose not to edit their learner model, compared to 7 participants with previous OLM experience who chose to edit. Each of the three who did not edit, all stated that they believed the model to be an accurate reflection of their learning, so did not wish to change the learner model, “I did not know the answers so did not want to alter model as it was an accurate reflection of my C understanding” and “No, I did not want to edit, as the model showed me what I knew”.

3.7 Discussion

We first consider the erroneous data that was inserted into the LM. The erroneous data (misconception or knowledge) was placed on the extremes of the learner model scales (believe – not believe). This meant a maximum of 4 scale point moves could be made, and if the learner was to fully understand their learning, this would be the outcome of their editing. As in past research [9], results have shown that learners could, with some success edit the inserted misconception in the learner model, but not the inserted knowledge. In this previous research, editing has not been measured on a scale necessarily [9], here it was done using a drop down menu and where a scale has been used [9] it was smaller. So, any movements on the scale or changes to the LM would have been more significant than is available with the 5 point scale used in this research, which allows the participants to make smaller changes to their LM.

Of the 20 participants taking part in the study, 15 edited the misconception inserted into their learner model, of which 8 edited the full 4 scale point moves, making the full amount of scale point moves shows a certainty in what the learner is editing. If the learners were making just one scale point move, even in the correct direction, in terms of editing the inserted erroneous misconception, it is simply likely to be noise as a one point scale point move shows very little certainty in the edit. Therefore, even using a larger scale the learners showed positive accurate edits. This is different to the previous research [9] as in the past where misconceptions had been inserted, and participants had edited, they did so to the extreme of the scale that was available to them, whereas here not all participants did this. This could be down to the larger scale giving the participants a greater range (i.e. less influencing) to choose from. Of the five who did not edit the inserted misconception, three stated their reason for this as having had previous experience using OLM and believed their LM to be accurate, a theme common in past research [9 & 23]. The final two who did not edit their LM, had stated in the system and questionnaire that they were weak at both self assessment and C programming (issues discussed later), these could have been factors for them not attempting to edit the LM scale where the misconception had been placed.

Again, similar to past research [9], there were no participants who attempted to edit the knowledge erroneously inserted into their learner models, even with the larger scale offering the choice to make less impacting edits upon their LM. This could partly be the participant recognising what the learner model was showing them as their belief (inserted knowledge) as correct, but when answering the questions in the system, they did not have all the pre understanding required to get to that level of knowledge. This is an interesting point, as looking at it from another perspective, through the learner recognising the statement as correct they could in fact be learning. If the user (in a different scenario) was to then remember this knowledge and apply it in the appropriate manner, it could replace what they previously believed. Or it could be reasoned that if a participant recognises something as academically correct why would they change it, even if their belief is different? (For example people may know $E=mc^2$ but cannot explain and understand it, but may assume otherwise if a LM was to tell us we did understand it).

Editing the LM (erroneous data) has two very different outcomes, as 15 of the 20 participants made appropriate edits to the misconception placed in their learner models, of which the extents were: 40 % edited 4 scale points; 30% edited 3 scale points and 20% 2 scale points. None of the participants attempted to edit the inserted knowledge. The overall success of the editing is 35% (knowledge and misconception). This figure is relatively low, especially considering that it is the participant carrying out a self assessment. Success here meaning the participant made an edit where either the misconception or knowledge statements were inserted. The lack of appropriate/accurate editing could be down to an UK educational system which recognises itself, that very little self assessment is carried out throughout school, possibly affecting participants' ability to carry out an effective self assessment [14 & 26]. This result is supported by the participants who stated that they were weak at self assessment. This can be seen in the significance of the Chi-squared test, which indicates that there was a difference between what the participants believed their ability was and what it actually is. The test along with observing Table 2 shows that the participants did over estimate what they thought they knew, and in turn were actually weaker learners on the whole.

Next we consider the interaction with the learner model, and view how the participants edited their learner model, whether they made their learner models reflect that of a weaker or stronger learner in academic terms, and not what their actual beliefs were. Just because an edit makes a learner look stronger, does not necessarily indicate an accurate edit as the participant may have answered the questions showing a different belief to the one which is correct, and therefore the edit would be inappropriate for their learning.

Viewing Table 3, we can see how the participants edited their LMs in all 6 sections of the learner model, covering the system's domain. Of the 20 participants three chose not to edit their learner model. This may have been because of their previous experience with OLM, though 8 participants with previous OLM experience did edit their LM. The participants with experience using OLMs have all gained this from modules that are run within the School of Electronic, Electrical and Computer Engineering of the University of Birmingham. How much experience they had with OLMs was down to the participants own perception.

15 of the participants all edited their learner model to represent a stronger learner than they actually are, in academic terms. Just because the edits made them look stronger does not necessarily make them correct as some of the edits may not have been what they believed (as inferred from their answers), and would therefore be inappropriate edits. Three of these 15 participants made edits to individual concepts which made that particular section of their learner model weaker (academically), but

overall once all editing had been concluded, the LM model represented that of a stronger learner academically. It is interesting that learners' edits made themselves represent stronger learners than they actually are, as through use of the system they have demonstrated that they do not have the knowledge and that is reflected in their LM. Perhaps the learning material available for each section of the learner model could have helped learners to edit to what they believe is a stronger looking model academically, they could have learnt from the learning material. A potential drawback of static learning material is that the learner may have just taken the information from the learning material without really paying attention to it; this would mean the LM sections edited because of this would not be an accurate reflection of their learning.

An interesting point from the results is that 7 of the 17 participants who chose to edit altered nearly half of their learner model. This shows that participants were heavily involved with their learner models, and were responsive to the editable features. However, making such changes to the learner model can raise many questions such as; The ability of the learner? The use of the information pages? Were they editing to impress the researcher? Was the participant guessing? Editing so much of the learner model has implications for the learner, as the purpose of the learner model, a tool for learner guidance would be irrelevant as the LM would not represent the learner accurately. There could have been a larger number of edits due to the artificial setting of the research, as there is no outcome for the learner once they had finished taking part in the research. This could be a risk to the validity of results, as participants could have been making edits, knowing there were no consequences to their actions. How they then use the LM would become inaccurate as they would be using a LM which is not representative of their learning and furthermore, if a system were providing adaptive guidance (not static), this guidance could be based on an inaccurate LM. To prevent such issues, prompts could come up when a learner is making such large changes to their LM, reminding them of the implications of their actions, which could possibly encourage a more accurate LM.

Analysing the learners' ability in the domain (C programming), and comparing this against how the learners edited their LM, shows us firstly there were three participants that chose not to edit their learner model. Of the participants who did edit, two of them edited their learner model in a way that when they had finished editing their LMs, overall they reflected the same level of understanding. This is in terms of how many concepts within the domain showed a correct understanding. This implies that some of their editing portrayed a stronger learner, in some concepts of the learner model, and weaker learner in others. Regardless of their edits the model was still not accurate as the edits made to the model were not reflective of their learning, based on the answers they gave to the questions.

Out of the 20 participants 5 deemed themselves to have a very weak C programming ability, and their model contents also suggest this. The majority of the other 15 participants believed that they had a medium to weak ability at C programming. This is clear from the levels of understanding displayed by the learners once they had answered all questions in the system and taking into consideration what the models would have looked like prior to any editing and before inserting any errors. Another reason for the mixed results could be reflected in many participants' open ended questionnaire comments that they were not as good at C programming as they had originally thought.

When analysing the results there is an interesting point based on the number of edits one participant made. The participant in question has an actual learner model before editing reflecting a very weak learner, yet they made 16 out of a possible 24 edits to their learner model (24 edits maximum, with 6, 5 point scales, with a potential move of point 1 to point 5 equalling 4 scale point moves). More interestingly they were all edits demonstrating a far stronger learner than the participant had stated prior to using the system. However, overestimation is a common theme with weaker learners [31], where they will try and make themselves look better than they are, it could be argued though that the participant does not have a strong enough knowledge to judge themselves correctly rather than trying to make themselves look better. If this system was to be used and feedback was given based on the learner model, this could be detrimental to their learning, as the feedback would be based on the edits the learner makes and not based on their actual level of knowledge. Looking at the other extreme, one learner stated a strong ability at C programming and only made 8 edits, which also include 4 for a successful edit of the inserted misconception. This more cautious attitude about their learning and ability to self assess is common with stronger learners as they tend to know more about themselves [19 & 33].

11 participants made more than 8 edits, considering none of the participants edited the inserted knowledge. This indicates that participants were also editing parts of their learner model which were an accurate reflection of their learning. This implies that the edits the participants were making were not accurate and, if anything, could be detrimental to their learning. This is because as mentioned earlier, if the system provided feedback, this feedback would not be of use as it would be aimed at the edits the learner made and not their learning. Participants editing sections of their LM that is an accurate reflection of their learning would suggest they do not believe what they believe or they do not have the knowledge they say they have – more likely. It would be interesting to consider different ways of presenting a LM (scale, graphs, belief statement) to groups of participants and see if there are differences.

When considering the extent to which the learner will edit their learner model, including the optional learning material to view, would hopefully prompt additional interaction. There were six learning material buttons, one for each concept within the domain. Each button presents the learner with some additional learning material to read on the selected concept of the learner model. This was a non adaptive form of learning material available to all of the learners. Only 3 of the 20 participants chose not to view any of the learning material. When considering the other 17 participants and how they edited their LM, it is important to consider that only two did not edit their learner model, and both of these participants had mentioned they believed their learner model was accurate. This could mean that some of the editing could have been influenced by the learning material, as learning material information may have prompted the learner, and could have then influenced the learner into making an edit as they now did not believe what was being shown to them in their learner model.

Two of the 17 participants that viewed the learning material, were not able to or chose not to edit the inserted misconception in their learner model. So even though they had shown they had knowledge of this section in answering the questions (the misconception was erroneous) and viewed the learning material for this concept they still chose not to edit their learner model. This could mean the participant was guessing their answers and did not understand the concept, or they believed what was being shown to them was correct. Considering the learning material, on the whole participants edited in a way that made their learner model reflect that of a stronger learner in terms of academic correctness but not necessarily in terms of what they actually believed based on the answers they gave to the questions in the system. However, as stated above viewing the materials may have been sufficient to prompt recognition.

Two of the participants who edited their LM and viewed the learning material as mentioned previously made edits, that when completed still made them reflect the same level of understanding they had demonstrated before choosing to edit. The remaining 13 participants who viewed the learning material all made improvements to their learner model with respect to editing the inserted misconception in their learner model, this could have been influenced by the learning material. However the same argument could be made in reverse that none of the participants were able to edit the inserted knowledge. This could have been because the learner viewed the learning material and then recognised that the LM was in fact showing the correct understanding of the topic, so they would not need to edit their LM.

An interesting point here is that there seems to be no relationship between viewing learning material and total edits made (correct or incorrect), as both sets of participants who chose to view or not to view the learning material have similar ranges, medians and means. This wide range and respective medians and means on both sets of participants would suggest that viewing the learning material could be serving multiple purposes: viewing it to try and correct their learning or using it to back up their own beliefs, this could be where the learner views the learning material but does not edit their LM. To state this clearly and conclude it further research would be required as there were not enough participants who did not view the learning material.

Relationships of interest with regard to viewing the learning material and making an edit to the learner model are that 5 participants chose to view all of the learning materials, however two of these were participants chose not to edit their LM, and this was because of previous LM experience as stated in their questionnaires. So here the participants could have just been doing further reading on the concepts, possibly building their knowledge. So the point here is that even though the participants have viewed the learning material and possibly increased their understanding of the concept, they still chose not to edit, it would have been useful to also ask participants why and what for they used the learning material, not just how useful they found it. Two of these participants were participants who chose not to edit their learner model, the other participants may have believed the LM was accurate and chose not to edit their LM.

The learning material was quite extensively used by the participants, and the post research questionnaire just asks about its usefulness, not why it was used. It would have been useful to have asked the participants about why they used it so much, as it could have given the discussion some insight into the participants' use of the learning material and not just trying to offer answers based on the observed behaviour of the participants.

15 participants made edits where they had viewed learning material and of these 15 participants, 4 only edited sections of their LM where they had viewed the learning material. This is interesting as it suggests that 20% of the participants have used the learning material as an aid in whether or not to make an edit to their LM and not actually what they believe. Alternatively, here the participant could have now understood more about a concept and therefore changed their learner model to what they now believe the LM should be showing. However if they were just making these changes based on the learning material and not what they understand, potential feedback could be wrong and is it would be aimed at the wrong level for that learner.

When looking at which sections of the learning material were viewed, it is interesting to note that all participants viewing the learning material viewed the first section of the learning material. Looking further at this, there is heavy use of the learning material in the early sections (sections here is representing a concept) of the learner model with 13 participants viewing the first section of learning material, 12 viewing the second section of learning material and 8 participants viewing both sections, compared to 5 participants viewing the last sections learning material. This matches the participants' editing patterns. So the use of the learning material could have a direct effect on whether a participant edits a particular section of their LM. The usage decrease could have just been because of the time of the session.

There is a less manageable condition that needs mentioning purely because it could have some effect on the results as stated in previous literature [23 & 28]. The participant trying to impress the researcher, although the use of the system does not affect the participant after use, they still might want to use the system how they believe the researcher wants the system to be used. This type of interaction made by the learner could give some false results, as the learner will not be showing what they believe but what they think the researcher wants. To minimise this risk the instructions were as to the point as possible and reassuring to the participant that no results would be attachable to them and only used for the purpose of the research.

Two issues which can be cross analysed throughout the discussion are self assessment and C programming (strength of learner) ability. These two points could be considered two of the most important and can be considered alongside all other topics and help explain why a learner may have acted as they did. These two areas are pivotal, firstly because ability will affect how well the learner can answer the questions, thus potentially affecting how they interact with the model. Secondly the ability of the learner to self assess is crucial, because even if they are not good at the subject they should have some understanding of their own learning. These two factors could have great effect on the extent to which the learners edit their learner model.

Table 3 shows self assessment and C programming ability, against the system's learner model, the learner model including the erroneous data and the learner model after editing. The three participants who did not edit their learner model all had mentioned they had previous OLM experience as to why they were not editing, but they also stated that they were weak at C programming and self assessment. This judgement of themselves could have affected their choice of not to edit.

Considering the level of C programming, all remaining 17 participants chose the middle levels of skill on the 5 point scale (Very strong – Very weak). Viewing the participants' learner models in the system prior to editing, shows that the participants had quite accurate reflections about themselves. This middle rating could be due to the participants not wanting to state either way and could be a reason why some of them stated in the questionnaire they knew less than they realised. It would be reasonable to suggest having a gauge of some description, to judge the certainty of the answers the participant is giving for each answer. By doing this it could be possible to reflect more accurately how well the participant actually knew the concepts within the domain. This would eliminate the uncertainty of the participant's believed ability for the whole domain, as each question would have a certainty value, which would build a question by question profile of their perceived ability in relation to the actual answers they are giving. This could potentially give more insight into individual answer, concepts and domain ability in relation to perceived ability, and how this could change question by question, to concept and the domain.

Two of the participants after editing had made their learner models represent the same level of overall understanding, through editing some sections to suggest a stronger learner and others to show a weaker learner, so overall the model demonstrated the same level of understanding but in different sections of the domain to what the model initially looked like. This would be quite representative of their learning ability as they were unsure about certain topics and their edits suggested this. This could be why they made inappropriate edits by altering sections where no erroneous data had been inserted. This ability to have a model representative of the same learning ability, but with knowledge represented in different sections could have been down to a good self assessment ability, which both participants said they had.

The remaining 15 participants however all made edits which improved their learner models, making their LM reflect that of a stronger learner. Five of these participants had made their learner models represent a very strong learner. What is interesting about this is that all 5 of these participants had the highest opinions about their C programming ability. Although making their models stronger, they did not make that many inappropriate edits as they already had good learner models, and one of the sections they improved was the inserted misconception.

The participants who stated lower levels of C programming ability were far more adventurous in their editing; making much bigger changes away from what their learner model should look like, so going from a very weak learner to a strong learner. This is in line with previous work stating that weaker learners tend to overestimate what they think they know [34]. This is most evident in the one participant mentioned earlier with a very weak understanding of C programming who actually made the most edits to their learner model. Viewing the results of the Chi-squared significance test in table 5, it is clear to see that there is a significant difference between what the participants believed their model should look like (expected value) and what it did look like (actual value, post editing). Considering this result and the other Chi-squared result, which indicates the learners were weaker than they believed, would agree with previous research in that weaker learners do over estimate what they think they know. We can see this in the way the participants edited their learner models – by making them look stronger than they are.

7 of the 20 participants considered themselves to be good at self assessment. Of the 7 participants, 5 from this group edited the inserted misconception; however, the two participants who did not edit the misconception did not edit their learner model at all. Of the same 7 participants, three participants kept their learner models the same, two of these did not edit, and one made edits that although changed the model (but inaccurately), still represented the same level of understanding of the domain, but with their knowledge represented in other areas. Two of the participants showed very little differentiation from how their learner model looked after editing to what the system's representation is, not including the erroneous data. However, two of the

participants showed large variations from what their learner model looked like after editing, to what the system's representation of their learning was. Both of these participants had stated they were weak at C programming. This tendency of weak learners to overestimate what they know could probably be the reason behind the large changes.

The remaining 13 participants, when considering their self assessment skills, had all been cautious and stated a weak ability in self assessment. This could be due to a UK education system that realises there is very little formal self assessment carried out by students whilst at school level [14]. This could possibly be why so many participants stated a weaker ability and made lots of edits. Having the chance to self assess could have prompted these participants to interact more with their learner model, given that it is something that does not normally happen.

The ability of the learner and their self assessment skills, has shown similar patterns to previous research [9], and do support the reasons for why some learners interacted as they did. The study has shown that some participants will heavily edit their learner models, while some do not edit their learner model at all. In this study there has been a larger scale which has given the learner the opportunity to make less extreme edits, unlike the previous research, this opportunity was used as not all edits were the full extent of the scale. Having such a variance of learner patterns could be down to many factors: overall ability, ability in a particular topic within a domain, self assessment skills or use of the learning material.

Based on the results and the interaction of the learners it would be appropriate to suggest using an editable learner model making the learners aware of their editing actions and the effects this could have on the outcome of any feedback they may receive from the system (if developed in an ITS) or from an expert in the area of interest.

The results also showed us that the stronger learners made less edits to their learner model, where as the weaker learners made more extreme edits. An Interesting result in this research compared to previous research is the editing of the misconception with only 40% of participants who chose to edit, editing the full extent of the scale. The other 70% edited either 3 or 2 scale point movements, which could show us that having the larger scale lets the learners make more cautious editing decisions than previous work.

When carrying out research it is important to consider and analyse any possible limitations of the study. The first one to consider for this research would be the questions on C programming, if they were incorrect this would jeopardise the results and findings. The questions were designed based on a module ran with department of Electrical, Electronic and Computer Engineering. A concern that could have indicated a problem with the questions is if the participants had commented on any of them, especially if it were multiple times, this was not the case.

Another limitation could have been the how the participants edited their learner model and if there was excessive editing in particular sections. The results would indicate there was more editing towards the start of the system use, but this would coincide with where the inserted misconception was. The levels of editing in other sections of the learner model differed a lot between the different participants, so the results would suggest there were no problems with the learner model and questions, and none were raised in the additional comments section of the questionnaire.

Chapter 4 – Conclusions and Future Work

4.1 Conclusion

The research “Accuracy of a learner’s edit, to their learner model” was considered in relation to how appropriate the edits of the participants were and the number of edits the participants made in relation to the erroneous data and other parts of their LM. The research investigated how participants interacted with their learner model and what the outcomes were from these interactions. The system was developed around the domain of C programming, designing questions based around 6 concepts within the domain, to cover the whole domain would have meant developing a much more complex open learner mode and far larger question base than was needed for this research. In one of the topics there was a misconception inserted and in a second topic a piece of knowledge.

Along with the accuracy of the edits, many other factors have to be considered that could have affected the participants and how they interacted with their LM: self assessment (SA); participants ability (based on their own opinion and what the system shows); OLM experience; learning material (information optionally available to the participants), extent of the edits (how many scale points moved). Along with this, the research also considered whether the edits made are in fact correct in academic terms (factually correct as opposed to correct for the learner’s learning), as this links in with ability and how the strength of a learner’s knowledge could have affected their editing.

As in past research, the results in this research followed the trend of the participants being able to successfully edit the inserted misconception but not the inserted knowledge. This suggests that learners can spot a misconception and successfully edit a misconception. The research here clearly demonstrates that participants are unable to edit inserted knowledge that is in their learner model. Even more noticeable is that participants were editing accurate parts of the learner model which were showing an accurate reflection of their learning. This research does show that participants are able to identify erroneous misconceptions that are placed in a learner model, modelling both knowledge and misconceptions, but is less clear when it comes to inserted knowledge.

The erroneous data is the main focus point of the research, and is a good measure of the extent to which the learner will edit their learner model. The ‘extent’ is linked to how many scale points the learner will change in their learner model on the two topics related to the erroneous data firstly, and any other edits the learner makes to their LM. What is interesting is the difference between the two sets of results when looking at the inserted misconception and knowledge separately. The difference is large, with 75% of the participants successfully editing accurately to some extent the inserted misconception with 40 % managing to edit the full extent (four scale point moves). This was in contrast to where the knowledge was inserted, where none of participants edited accurately.

This was interesting purely because the fact all of the participants had shown no knowledge of the topic where their inserted knowledge was in the questioning, yet did not edit their learner model to show this. This is interesting, as it could be simply argued how the participant would know to edit this section of the learner model as they have shown limited knowledge when answering questions on this topic in the system. The same point could be argued for where the misconception was inserted as by being able to edit the inserted misconception the participants is demonstrating the knowledge they showed in answering the questions.

The research did manage to find out the accuracy of which learners can edit their learner model, as the participants involved all interacted with the system and answered sufficient questions for the system to model the learners in all topics within the domain. The accuracy to which learners edit their learner model can be described as an ability to identify and correct inserted erroneous data; but also, as the results show, how many inappropriate edits the participants made to sections accurately reflecting their learning. The research showed that participants can successfully edit inserted misconceptions in a learner model which models both knowledge and misconceptions. On the other hand, no participants were able to accurately edit the inserted knowledge to any degree, even with a larger scale in the learner model, allowing for edits which would be less altering to the learner model.

It is this extreme difference between editing inserted misconceptions and knowledge and the fact that some learners edited some accurate parts of the model that makes the area truly interesting.   Considering the individuality of learners and possible developments and choices of editable learner models, it makes the possibility of further research very real.

4.2 Future Work

The system designed and implemented was an independent editable learner model, which offered static learning material and had two pieces of erroneous data inserted into it. One piece of erroneous data was a misconception and the other a piece of knowledge. The system was designed to research the participants' ability to accurately edit their learner model. Their LM comprised of a belief statement followed by a 5 point editable scale indicating the strength of that belief. This was achieved through the participants answering multiple choice questions on various concepts within the domain of C Programming. The accuracy of the edits was considered alongside ability of the participants, perceived self assessment skills of the participants, any previous OLM experience and the extent of the participants' edits.

A possible development of this particular research would have been to perform a double cross over study, by splitting the participants into two groups. The research could have used the same system as in this research on one occasion and then changed the order of the questions in the second system. Doing the study this way would have addressed some of the possible limitations of the research, by firstly having a different order to the questions, and the way the learner model is presented could possibly show a difference in the amount of editing in certain parts of the learner model. As this research showed a lot of editing in the first parts of the learner model, could this have been because of the earlier questions? By having the system shown to the learners in two groups and two orders and using both versions of the systems, this would address any individual variance from having the one system and having the same order of questions.

If a double cross over study was carried out, the erroneous data would have also been displayed at different times within the system. This may have also possibly affected the participants' ability to edit the erroneous data, although this research and previous research would suggest the participants would still be able to edit the inserted misconceptions. The main benefit of the double cross over study would be to reduce the limitations of the research in terms of presenting the questions in different orders along with the presentation of the learner model, this would possibly show any areas of common editing, for example in this research the first three sections of the learner model. If the same three sections are edited regardless of the order presented to the users on two different occasions it could be down to other reasons, for example the questions or the concept.

The findings of the research suggested that participants were able on the most part to accurately edit the inserted erroneous misconception, with no participants successfully editing the inserted erroneous knowledge. This might suggest that it is particularly useful to allow editing of misconceptions within an editable learner model, as the participants in this research show initially that it can be done accurately. Based on the results and findings this research suggests several directions for further work.

A way of taking this research further would be to consider the lack of apparent success at editing the knowledge inserted into the LM. There are various ways in which to research this, with the use of two groups of participants. The first group of participants could have a learner model with inserted knowledge, but the learner model does not model misconceptions. The idea here is that the research presented in the thesis showed no editing of the inserted knowledge even with a larger scale for editing, but could the inclusion of misconceptions have distracted the learners and possibly prevented them from editing the knowledge.

The second group of participants would again have a learner model with inserted erroneous knowledge but have a learner model which models accurately and does include any misconceptions the learners may have. How would the inclusion of the misconceptions affect the ability of the participants to edit the inserted knowledge, even though the misconceptions would be an accurate reflection of the learners learning?

The idea behind this research would be to see if learners can edit their learner models accurately when a learner model does not model misconceptions. By having the two groups of participants it will be clear to see if this can be done. As the research in this thesis and previous work [9] shows that participants cannot edit inserted knowledge, when there is also inserted misconceptions. This proposed further research would then give initial results into how accurately the participants might be able to edit inserted knowledge in both a scenario where a model does not model misconceptions and a model which does model misconceptions but only ones the learner does actually have. This research could then possibly clarify whether editable learner models would be worthwhile in a system that does not model misconceptions and only shows knowledge, as the ability of the participants to accurately edit would suggest this.

Suggesting a LM with just knowledge inserted as above, it would be useful to compare the results and editing patterns of the participants in terms of appropriate edits and extent of edits. This research showed some of the participants made edits in parts of the LM which were reflective of their learning. Having a LM with erroneous knowledge inserted into it, making the learner appear stronger could make the participants edit less, as the model is representing a stronger learner, and stronger learners tend to make less edits to their learner model. So, even though the participants could be making less edits, as the learner model looks stronger, they would still not be editing parts of the learner model which are not reflective of their learning, where the erroneous data is inserted. So where a model does not model misconceptions will interactions with the editable learner model be less?

Following on from the above suggestions and the fact some of the participants were editing accurate parts of the learner model, it would be interesting to research the difference in editing between participants who have a learner model where erroneous data has been inserted and a group with no erroneous data. Based on previous results it would be easy to suggest that participants would edit accurate parts of their LM. However, in the research where erroneous data has been inserted some participants have stated they edited parts of the LM if they disagreed with it. So, the fact some parts of the LM were not accurate could have influenced the learners, and possibly lead them into editing other parts of their LM, suggesting a possible lack of trust in the learner model. If however, participants were editing the LM where no erroneous data was been inserted it would be interesting to do extra analysis with the participants to find out exactly why they acted the way they did.

Seeing how many inappropriate edits learners made in these two groups of participants would be interesting. Would a completely accurate learner model still be edited by some of the participants? If the edits were made to an accurate learner model further research into type of OLM domain, type of participant and presentation of OLM could be considered. This research could work alongside the learner model research, with just knowledge inserted in it as it would help further identify the types of editable learner model that would be appropriate for use, and the results together would show the best type of editable learner model to be deployed based on which type of learner model indicated the most number of accurate edits or least amount of inaccurate edits (ideally none).

A final development, based on the results of this research could be where the placement of the erroneous data is. For example here and in previous research [9], the erroneous data has been placed on the extremes i.e strongly agree or strongly disagree, with the belief shown in the participants learner model. If the erroneous data was placed on less of an extreme, how would this affect the editing of the learner model? Would the small increments in erroneous data affect success of editing a misconception? Would there be any success in editing inserted knowledge?

Results gained would be a direct progression from those in this research. The interesting development here would be the smaller differentiation between the erroneous data and what the LM should look like, how extreme the erroneous data was. Would this research impact on the ability to accurately edit misconception, probably not as in this research of the participants who did edit the misconception 75%, 80% of them edit the full extent of the scale. Again this would probably be more useful in determining the success of editing inserted knowledge into the learner model. It could help identify when learners may be willing to make edits to inserted knowledge, because if the erroneous data was less extreme, the edits the learner makes to the model will be less extreme as a result of this.

A final point on this development would be how less extreme erroneous data affects the extent of edits the learners could make on accurate parts of the LM. Would less extreme erroneous data mean less editing on accurate parts of the model? Would the learners still edit other parts of their LM? Would other edits follow a similar pattern to the edits the participants make to where the erroneous data is inserted? Would a difference in how extreme the erroneous data is affect the extent of which a learner will edit their learner model?

The main focus of the future research would be to look into participants' ability to edit the inserted knowledge in an editable learner model. Researching this will possibly identify whether it would be worthwhile having an editable learner model that does not model misconceptions and would participants be able to successfully interact with it. Furthermore the suggestions will help identify any possible reasons why participants seem to struggle with the inserted knowledge, is it because of the distractions of misconceptions or how the erroneous data in this research was placed on the extremes of the scale?

References

(1) Ahmad, N. and Bull, S. (2009). Learner Trust in Learner Model Externalisations, in V. Dimitrova, R. Mizoguchi, B. du Boulay & A. Graesser (eds), Artificial Intelligence in Education 2009, IOS Press, Amsterdam, 617-619.

(2) Beck, J., Stern, M. and Haugsjaa, E. (1996). Applications of AI in Education. ACM Crossroads. Special issue on artificial intelligence, 3(1), 11-15.

(3) du Boulay, B. Can We Learn From ITSs? 2000.

(4) Bull, S. 2004. Supporting Learning with Open Learner Models.. 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education, Athens.

(5) Bull, S and McKay, M. 2004. An Open Learner Model for Children and Teachers: Inspecting Knowledge Level of Individuals and Peers Intelligent Tutoring Systems: 7th International Conference.

(6) Bull, S. and Mabbott, A. (2006). 20000 Inspections of a Domain-Independent Open Learner Model with Individual and Comparison Views, in M. Ikeda, K. Ashley & T-W. Chan (eds), Intelligent Tutoring Systems: 8th International Conference, Springer-Verlag, Berlin Heidelberg, 422-432.

(7) Bull, S, Quigley, S and Mabbott, A. 2006. Computer-Based Formative Assessment to Promote Reflection and Learner Autonomy. Engineering Education: Journal of the Higher Education Academy Engineering Subject Centre. Volume 1, No 1

(8) Bull, S., McEvoy, A.T. and Reid, E. (2003). Learner Models to Promote Reflection in Combined Desktop PC/Mobile Intelligent Learning Environments, in S. Bull, P. Brna & V. Dimitrova (eds), Proceedings of Workshop on Learner Modelling for Reflection, Supplemental Proceedings Volume 5, International Conference on Artificial Intelligence in Education 2003, University of Sydney, 199-208.

(9) Bull, S., Dong, X, Britland, M. & Guo, Y. (2008). Can Students Edit their Learner Model Appropriately?, in B.P. Woolf, E. Aimeur, R. Nkambou & S. Lajoie (eds), Intelligent Tutoring Systems: 9th International Conference, Springer-Verlag, Berlin Heidelberg, 674-676.

(10) Bull, S. and Pain, H.: "Did I Say What I Think I Said, and Do You Agree With Me?": Inspecting and Questioning the Student Model. Proceedings of World Conference on Artificial Intelligence in Education, Charlottesville, VA (1995) 501-508

(11) Bull, S. & Nghiem, T. (2002). Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers and Instructors, in P. Brna & V. Dimitrova (eds), Proceedings of Workshop on Individual and Group Modelling Methods that Help Learners Understand Themselves, International Conference on Intelligent Tutoring Systems 2002, 5-13.

(12) Bull, S. & Kay, J. (2007). Student Models that Invite the Learner In: The SMILI Open Learner Modelling Framework, International Journal of Artificial Intelligence in Education 17(2), 89-120.

(13) Zapata-Rivera, J-D. & Greer, J.E. (2001). Externalising Learner Modelling Representations, Proceedings of Workshop on External Representations of AIED: Multiple Forms and Multiple Roles, International Conference on Artificial Intelligence in Education 2001, 71-76.

(14) Karen Hinnett, Judith Thomas. Staff Guide to self assessment and peer assessment, 1999. Oxford Brookes University. Oxford Centre for Staff and Learning Development

(15) Holt, P and Dubs, S and Jones, M and Greer, J. The state of student modelling. Student modelling: The key to individualized knowledge-based instruction. (1994). Volume 125, pages 3-35.

(16) Kay, J.: Learner Know Thyself: Student Models to Give Learner Control and Responsibility. Proc. of Intl. Conference on Computers in Education, Kuching, Malaysia (1997) 18-26

(17) Judy Kay. The um toolkit for cooperative user modelling, 1995. User modelling and User adapted interaction. Volume 11,Issue 1-2, Pages: 111 - 127

(18) Kay, J. (2000). Stereotypes, student models and scrutability. In K. V. G.Gauthier, Claude Frasson (Ed.), Intelligent tutor systems, 5th international conference, its 2000

(19) Kay, J. Lichao, L. and Fekete, A. Learner Reflection in Student Self-Assessment. 2005. Proceedings of the ninth Australasian conference on Computing education - Volume 66, pages 89 – 95.

(20) Kerly, A., Hall, P. and Bull, S. (2006). Bringing Chatbots into Education: Towards Natural Language Negotiation of Open Learner Models, in R. Ellis, T. Allen & A. Tuson (eds), Applications and Innovations in Intelligent Systems XIV - Proceedings of AI-2006, 26th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer

(21) Kerly, A., Ellis, R. and Bull, S. (2007). CALMsystem: A Conversational Agent for Learner Modelling, in R. Ellis, T. Allen & M. Petridis (eds), Applications and Innovations in Intelligent Systems XV - Proceedings of AI-2007, 27th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Springer Verlag, 81-102. Selected as one of Best Application Papers

(22) Likert, Rensis (1932). A Technique for the Measurement of Attitudes Archives of Psychology 140: 1–55.

(23) Mabbott, A. and Bull, S. (2006). Student Preferences for Editing, Persuading and Negotiating the Open Learner Model, in M. Ikeda, K. Ashley & T-W. Chan (eds), Intelligent Tutoring Systems: 8th International Conference, Springer-Verlag, Berlin Heidelberg, 481-490.

(24) Mitrovic, A and Martin, B. 2002. Evaluating the effects of open student models on learning. Shared international conference on adaptive hypermedia and adaptive web based systems

(25) David Nicol, Debra MacFarlane – Dick. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice, 2006. Published in Studies in Higher Education

(26) Self, J. Student models in computer-aided instruction, International Journal of Man- Machine Studies, 6, 261-276, 1974.

(27) Self.J The defining characteristics of intelligent tutoring systems research: ITSs care, precisely, 1999. International Journal of Artificial Intelligence in Education

(28) Sleeman, D.H and Brown J.S, eds. Intelligent tutoring systems. 1982. Academic Press, London

(29) Tanimot, S. Dimensions of Transparency in Open Learner Models, 2005. Int'l Workshop on Learner Modelling for Reflection, to Support Learner Control, Metacognition and Improved Communication. Between Teachers and Learners.

(30) Tastle, W,J. Russell, J and Wierman, M. A new measure to analyze student performance using the Likert Scale, 2005. Information Systems Education Journal, 6 (35)

(28, 31) Teixeira, C.   Labid, S.   Nascimento, E. Modeling the cooperative learner based on it's actions and interactions within a teaching-learning session. - Frontiers in Education, 2002. FIE 2002. 32nd Annual

(32) Urban – Lurrain, M. Intelligent tutoring systems: An historic review in the context of the development of artificial intelligence and educational psychology, 1996

(33) Virtanen, P. Niemi, H. Nevgi, A. Raehalme, O and Launonen, A. Towards strategic learning skills through self-assessment and tutoring in web-based environment, 2003. European Conference on Educational Research

(34) Wade, A. Abrami, P.C, and Sclater, J. Journal of learning and technology. 2005, Volume 31(3)

Appendix

Participant Instructions

What you are doing:

You are about to take part in my MRes project

What you will be using:

You will be asked to use a computer based system where you will answer multiple choice questions on concepts within the domain of C programming. After a certain number of questions there is the option of viewing an editable learner model which will be displaying your beliefs based on the answers you gave to the questions. Once finished you will be asked to fill in a short questionnaire.

Your Data:

The first part of the system is a section where you will fill in a quick personal information page, but instead of filling in your name in the participant field, you will be given a participant number as instructed by the project investigator.

Ethical Issues:

- You may gain feedback once the experiment has been completely finished.
- All your information will remain anonymous, and with permission will be used for analysis and only be used by the project investigator/ supervisor
- You have the right to terminate participation in the experiment if you feel you need to do so.
- Any data gained in the experiment will only be stored for as long as needs be.
- The project investigator will be in the same room, so should you feel the need to ask any questions you are able to do so.
- The study complies with the British psychological standards for studies with human participants

Name:

Email: (optional):

I understand my rights as a participant in the experiment [ ]

I optionally chose to take part in the experiment [ ]

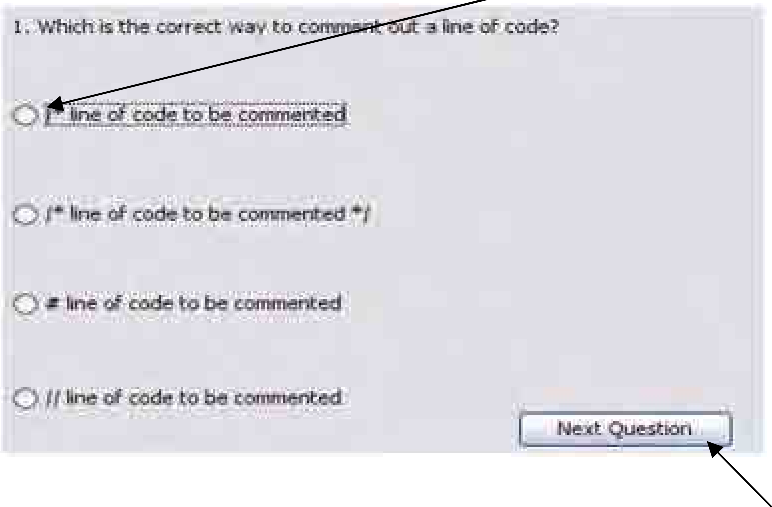
I give permission for my data to be used for research purposes only [ ]

I would like to be given feedback on the research [ ]

I would again like to thank you for taking part in my experiment.

A TYPICAL QUESTION SCREEN WITHIN THE SYSTEM

Select an answer by pressing on the radio button, next to the answer



1. Which is the correct way to comment out a line of code?

○ /* line of code to be commented

○ /* line of code to be commented */

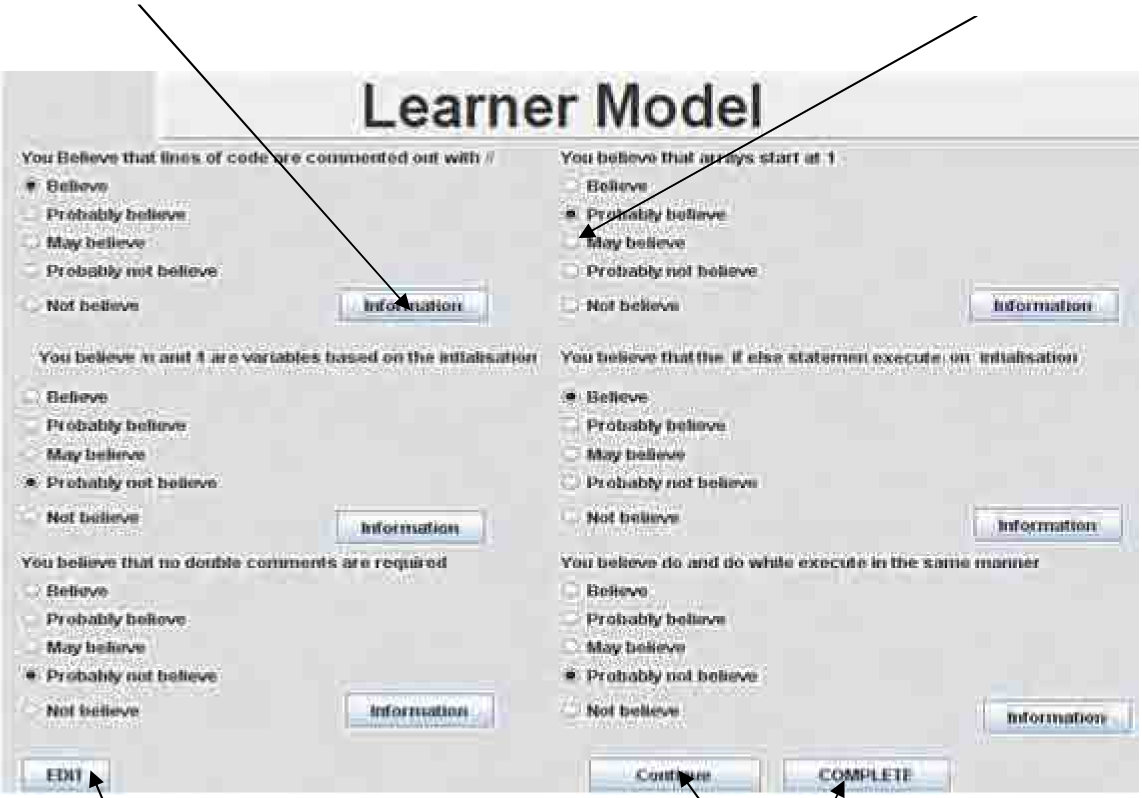
○ # line of code to be commented

○ // line of code to be commented

Once question is selected, press the next question button, or view model when available

Press to view information on the domain concept

Editable section of the learner model



Learner Model

You Believe that lines of code are commented out with //

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe that arrays start at 1

○ Believe
● Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe n and 1 are variables based on the initialisation

○ Believe
○ Probably believe
○ May believe
● Probably not believe
○ Not believe

You believe that the if else statemen execute on initialisation

● Believe
○ Probably believe
○ May believe
○ Probably not believe
○ Not believe

You believe that no double comments are required

○ Believe
○ Probably believe
○ May believe
● Probably not believe
○ Not believe

You believe do and do while execute in the same manner

○ Believe
○ Probably believe
○ May believe
● Probably not believe
○ Not believe

Press this button if you want to edit, it then highlights what concepts are available to edit

Press continue to continue answering questions. When all section of the LM have information in them click complete to finish using system

Evaluation Questionnaire

I would like to thank you for taking the time to use my system. I would now like you to take a few minutes to fill in the following questionnaire. All the data will remain totally anonymous and will only be used for evaluative purposes.

1. Participant number:

2. I have you previous experience with an OLM

strongly agree strongly disagree

5	4	3	2	1

3. I have previous experience using an editable learner model

strongly agree strongly disagree

5	4	3	2	1

4. I am good at self assessment

strongly agree strongly disagree

5	4	3	2	1

5. I viewed my learner model often

strongly agree strongly disagree

5	4	3	2	1

6. I viewed my learner model only when I had to

strongly agree strongly disagree

5	4	3	2	1

7. I viewed the additional information and it was helpful

strongly agree strongly disagree

5	4	3	2	1

8. I found editing my learner model useful

strongly agree strongly disagree

5	4	3	2	1

8.a Why/Why not

9. I believed my learner model was accurate

strongly agree strongly disagree

5	4	3	2	1

10. I edited my learner model when I disagreed with it

strongly agree strongly disagree

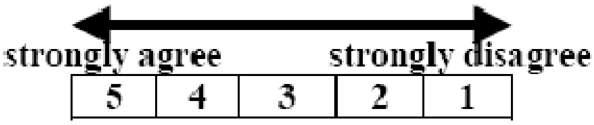
5	4	3	2	1

11. If my model showed a statement I did not agree with I viewed the learning material

strongly agree strongly disagree

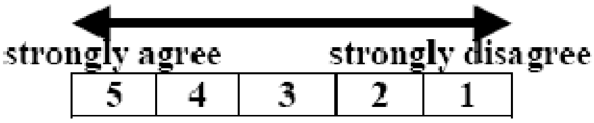
5	4	3	2	1

12. I understand the information in my learner model


strongly agree strongly disagree

5	4	3	2	1

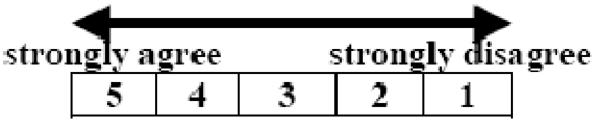
13. I would use an editable model again


strongly agree strongly disagree

5	4	3	2	1

13. a Why/Why not

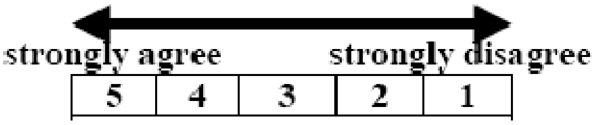
14. I found the editable model useful


strongly agree strongly disagree

5	4	3	2	1

14.a Why/ Why not

15. I found the experience of using the system useful

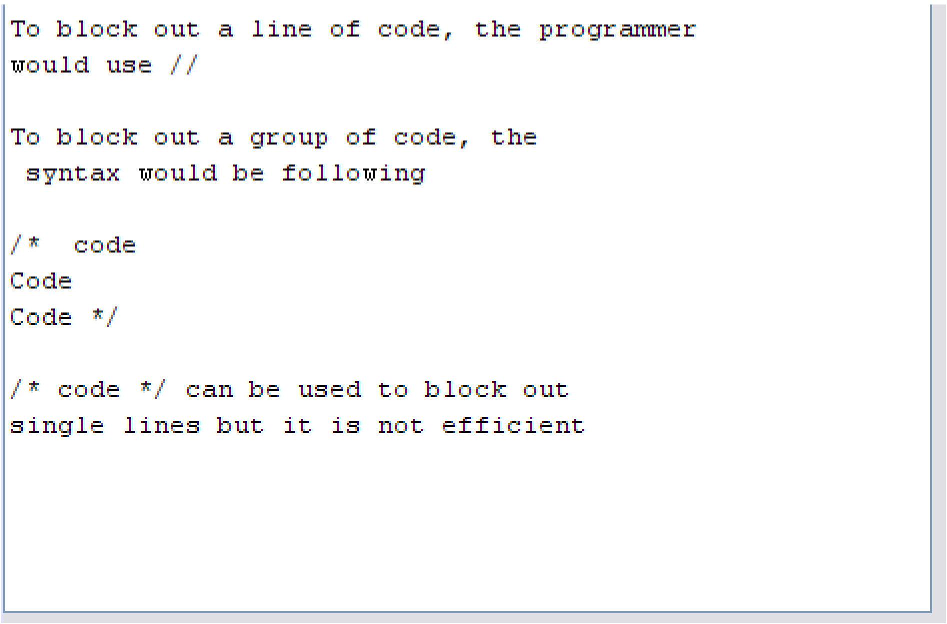

strongly agree strongly disagree

5	4	3	2	1

15.a Why/ Why not

16. Any additional comments

Information Screens

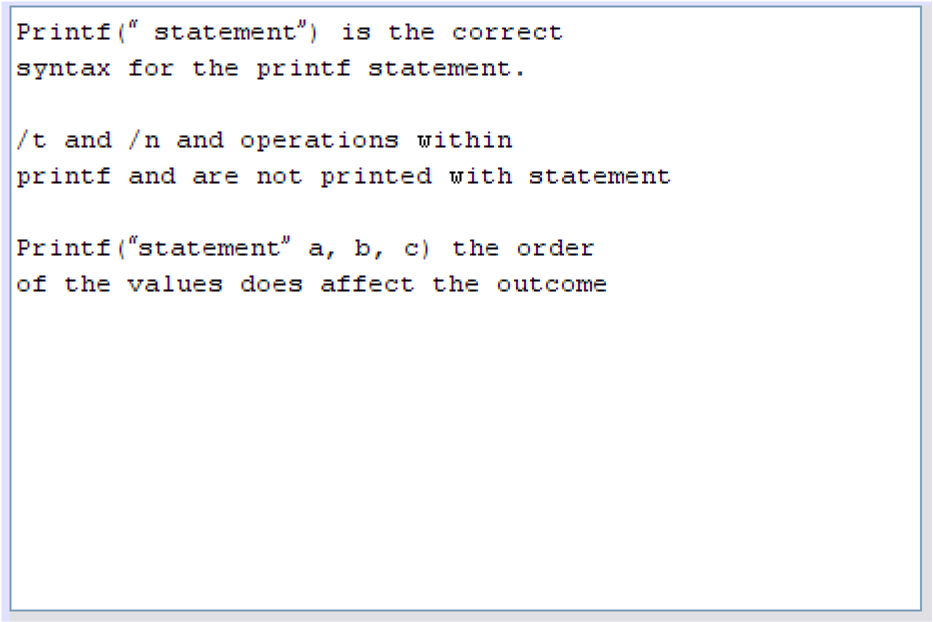


To block out a line of code, the programmer would use //

To block out a group of code, the syntax would be following

/* code
Code
Code */

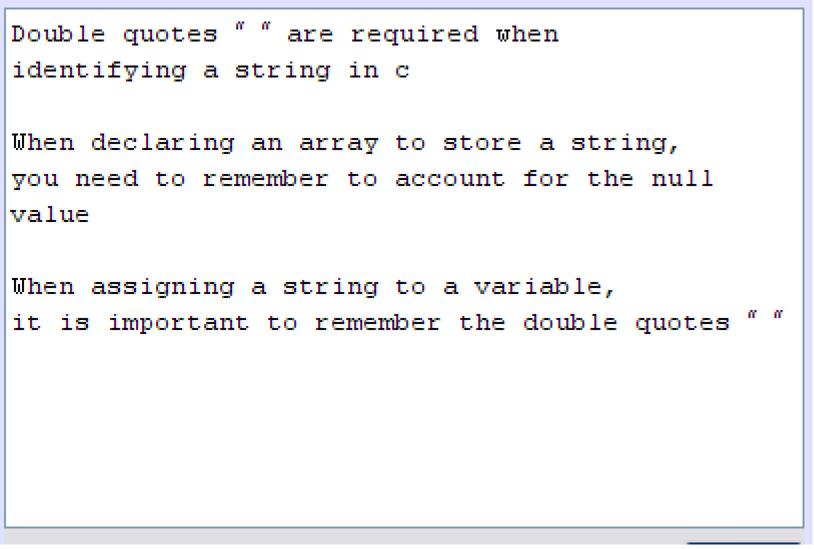
/* code */ can be used to block out single lines but it is not efficient



Printf(“ statement”) is the correct syntax for the printf statement.

/t and /n and operations within printf and are not printed with statement

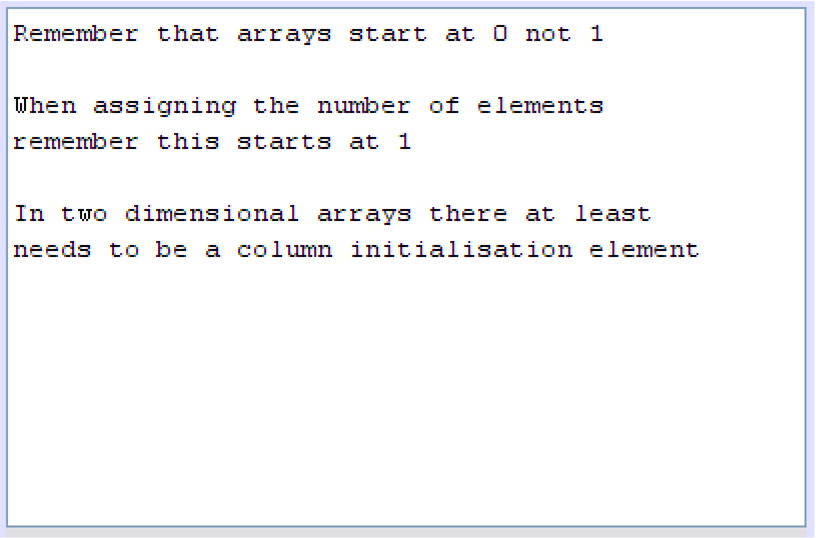
Printf(“statement” a, b, c) the order of the values does affect the outcome



Double quotes “ ” are required when identifying a string in c

When declaring an array to store a string, you need to remember to account for the null value

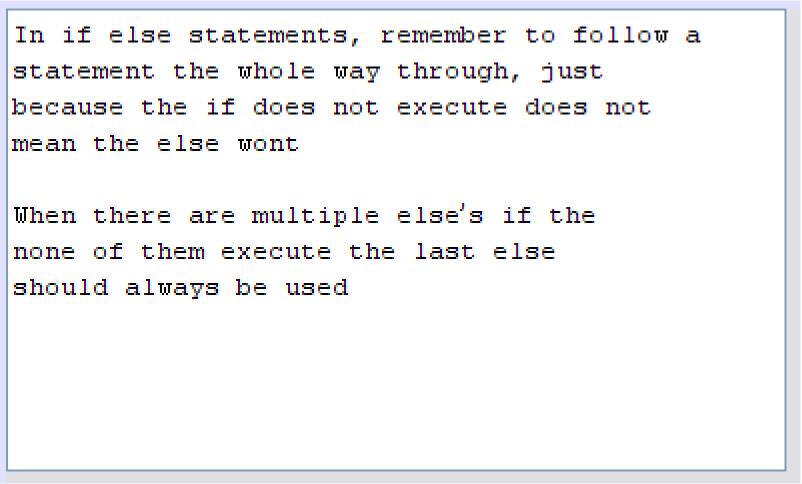
When assigning a string to a variable, it is important to remember the double quotes “ ”



Remember that arrays start at 0 not 1

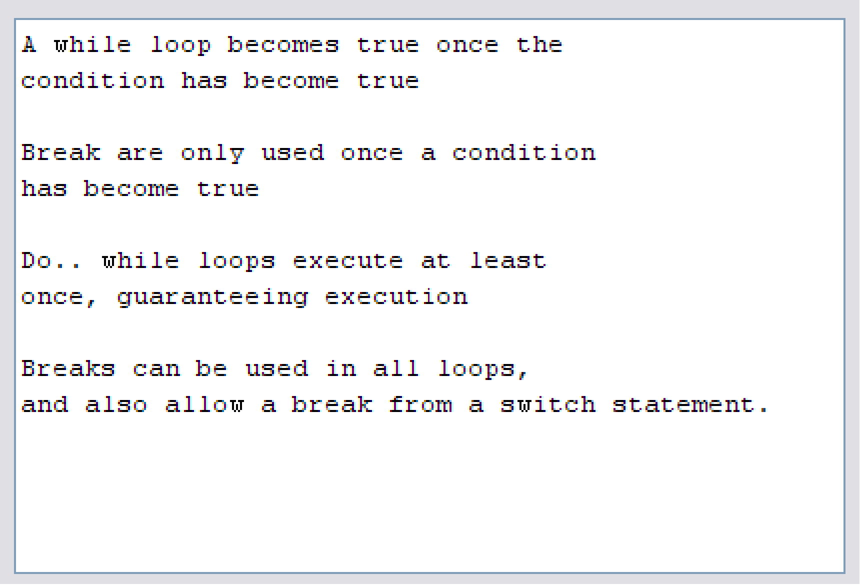
When assigning the number of elements remember this starts at 1

In two dimensional arrays there at least needs to be a column initialisation element



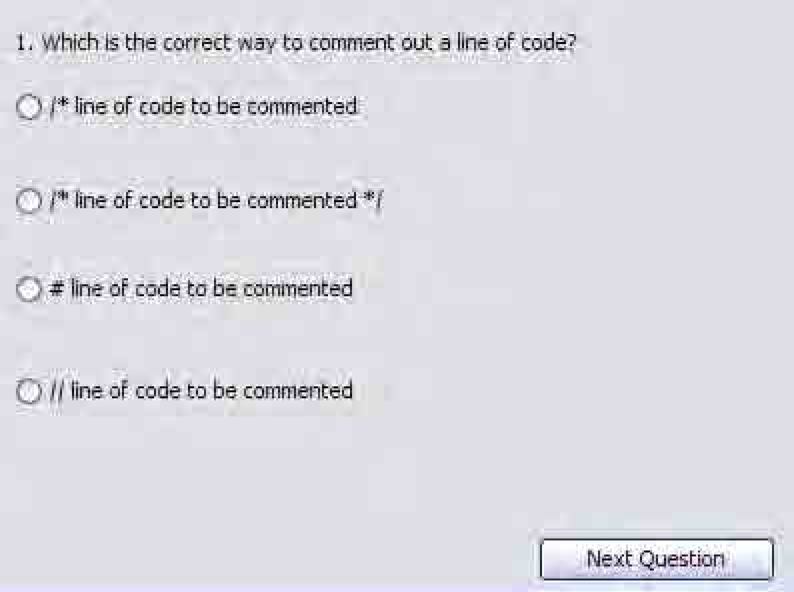
In if else statements, remember to follow a statement the whole way through, just because the if does not execute does not mean the else wont

When there are multiple else’s if the none of them execute the last else should always be used



```
A while loop becomes true once the
condition has become true

Break are only used once a condition
has become true

Do.. while loops execute at least
once, guaranteeing execution

Breaks can be used in all loops,
and also allow a break from a switch statement.
```

Question Screens



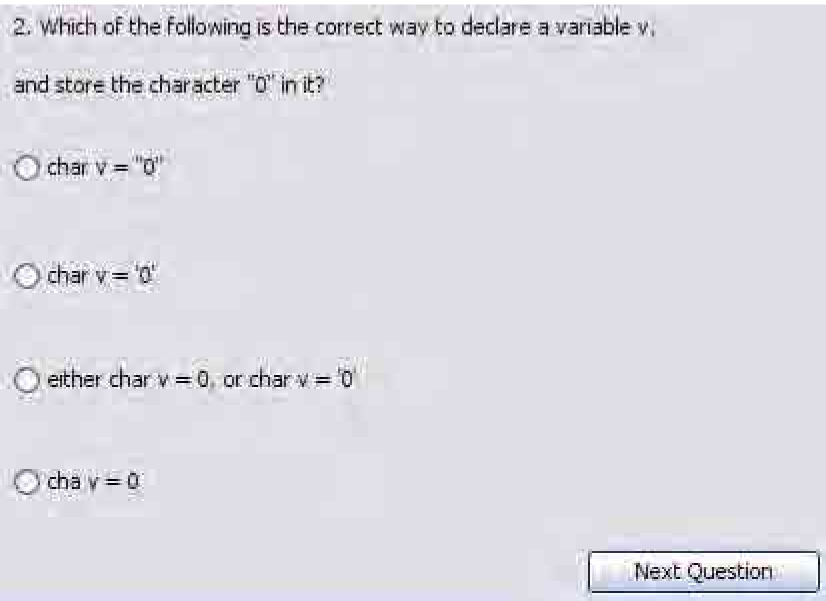
1. Which is the correct way to comment out a line of code?

○ /* line of code to be commented

○ /* line of code to be commented */

○ # line of code to be commented

○ // line of code to be commented



2. Which of the following is the correct way to declare a variable v,

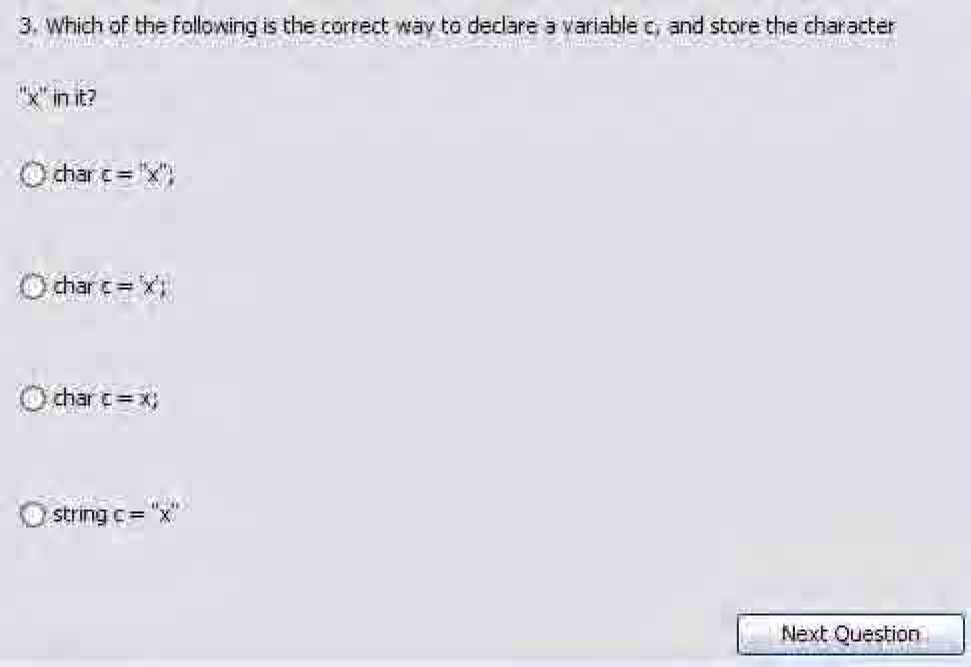
and store the character "0" in it?

○ char v = "0"

○ char v = '0'

○ either char v = 0, or char v = '0'

○ cha v = 0



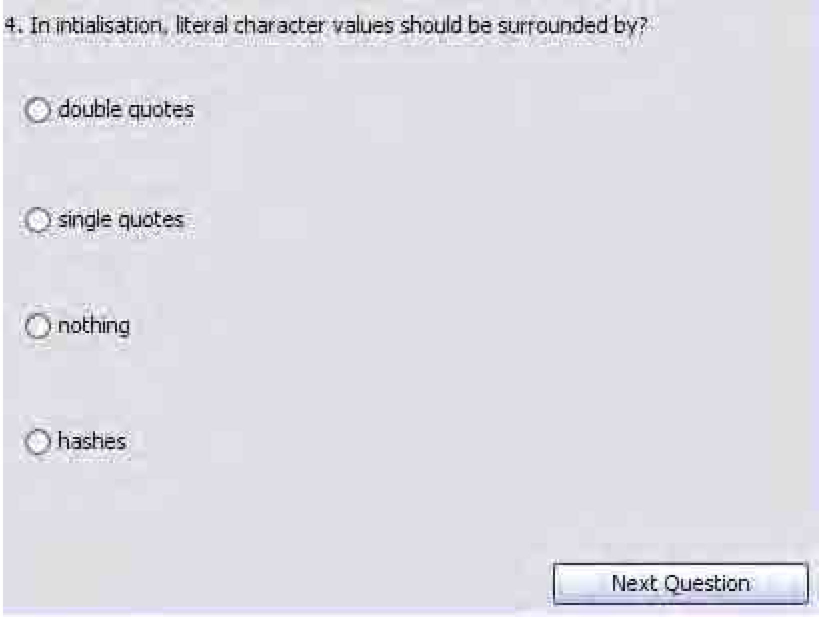
3. Which of the following is the correct way to declare a variable c, and store the character "x" in it?

○ char c = "x";

○ char c = 'x';

○ char c = x;

○ string c = "x"



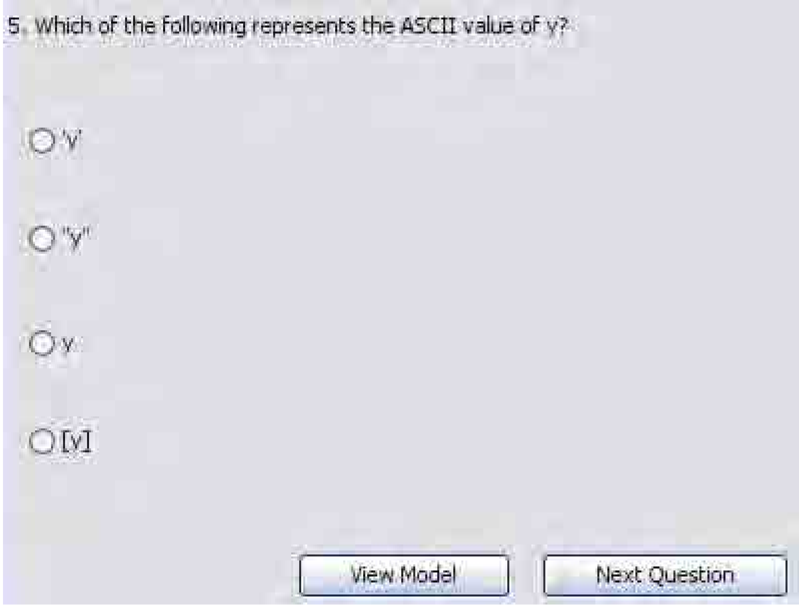
4. In intialisation, literal character values should be surrounded by?

○ double quotes

○ single quotes

○ nothing

○ hashes



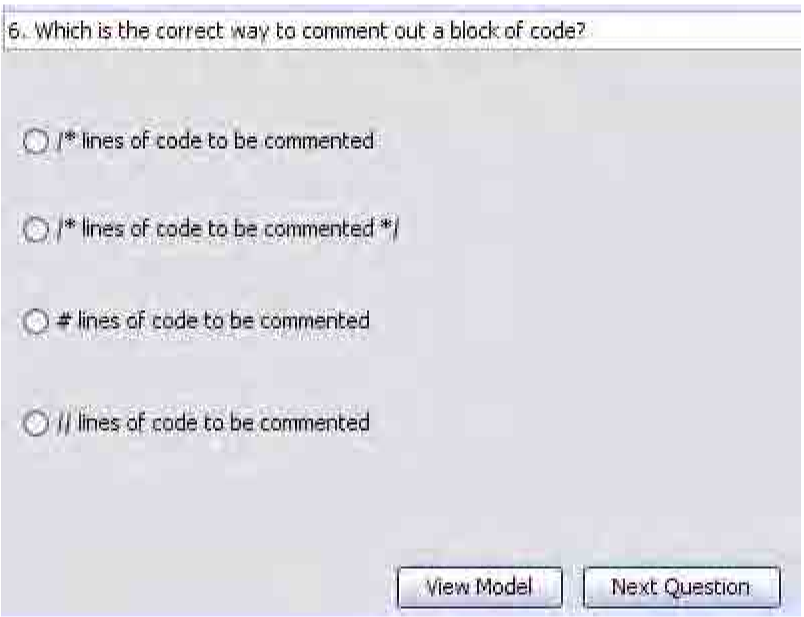
5. Which of the following represents the ASCII value of y?

○ 'y'

○ "y"

○ y

○ [y]



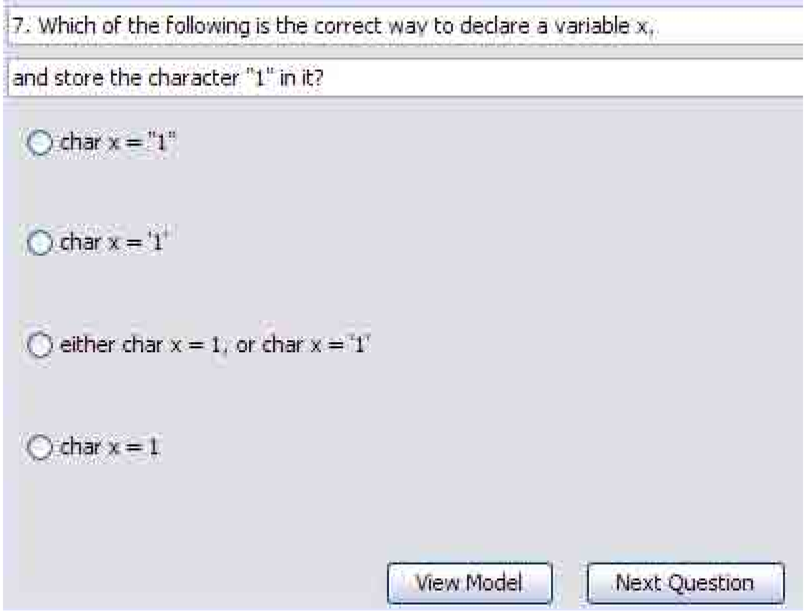
6. Which is the correct way to comment out a block of code?

○ /* lines of code to be commented

○ /* lines of code to be commented */

○ # lines of code to be commented

○ // lines of code to be commented



7. Which of the following is the correct way to declare a variable x,

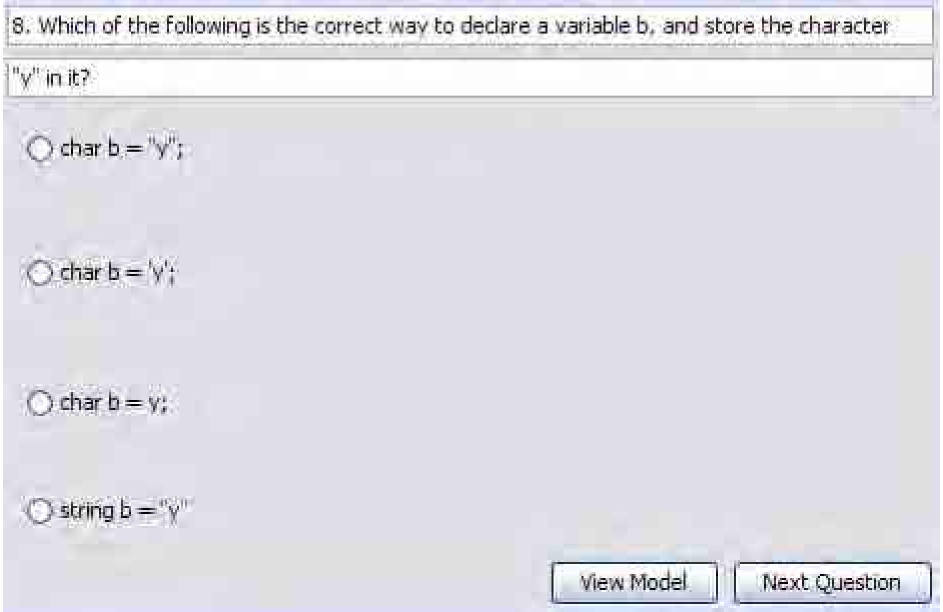
and store the character "1" in it?

○ char x = "1"

○ char x = '1'

○ either char x = 1, or char x = '1'

○ char x = 1



8. Which of the following is the correct way to declare a variable b, and store the character

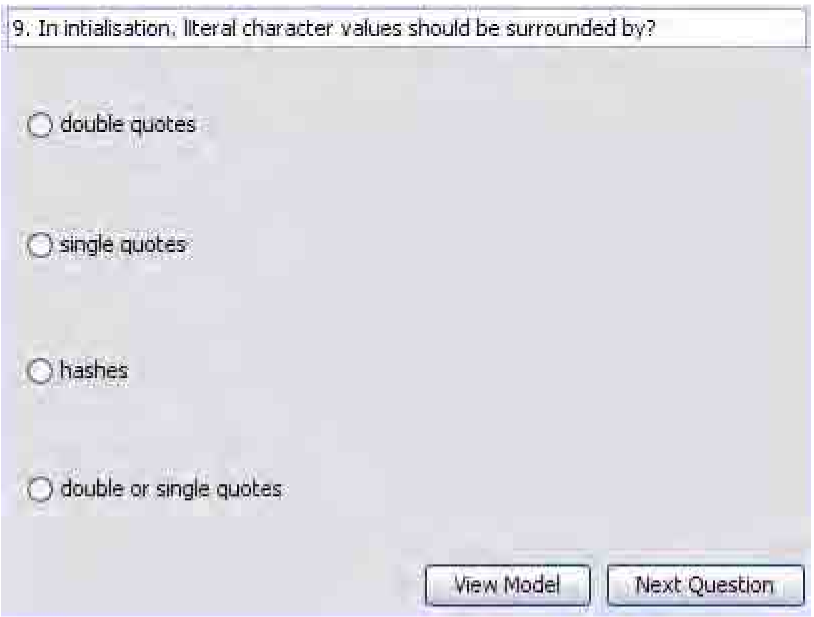
"y" in it?

○ char b = "y";

○ char b = 'y';

○ char b = y;

○ string b = "y"



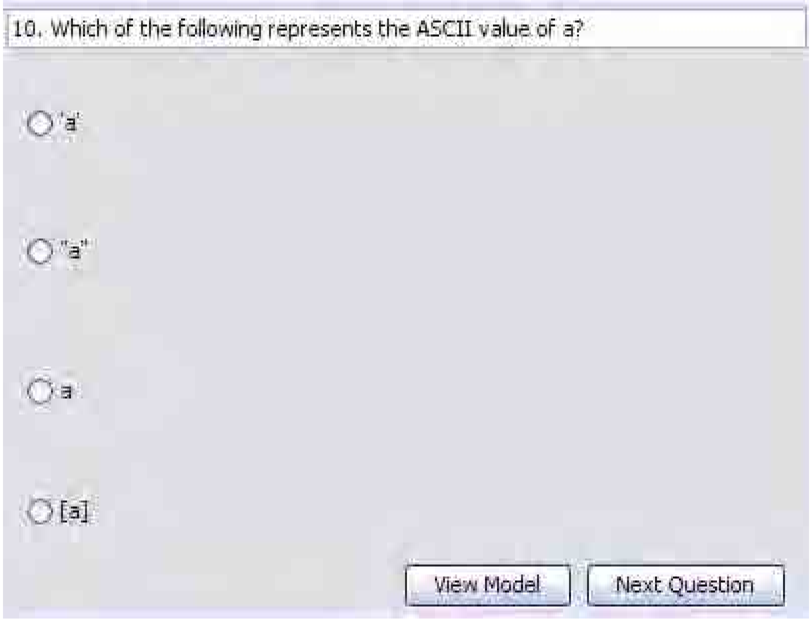
9. In intialisation, literal character values should be surrounded by?

○ double quotes

○ single quotes

○ hashes

○ double or single quotes



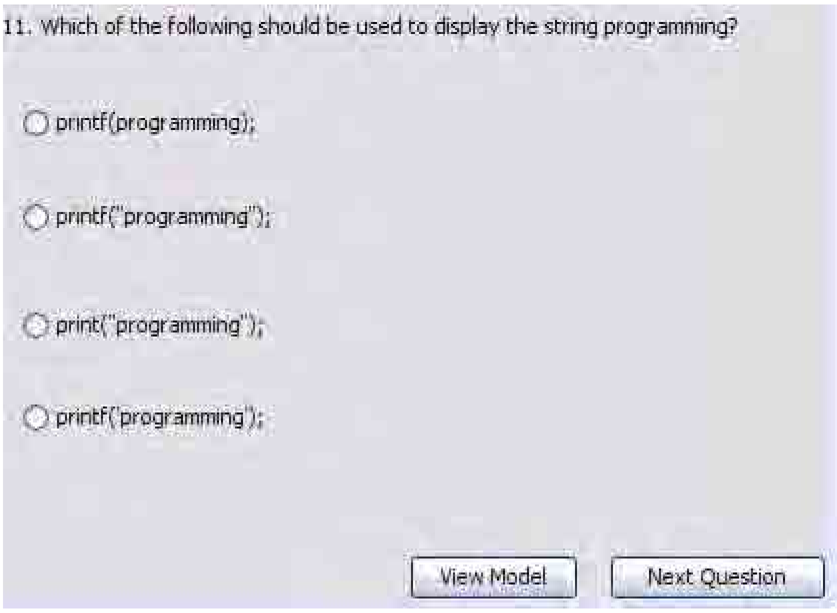
10. Which of the following represents the ASCII value of a?

○ 'a'

○ "a"

○ a

○ [a]



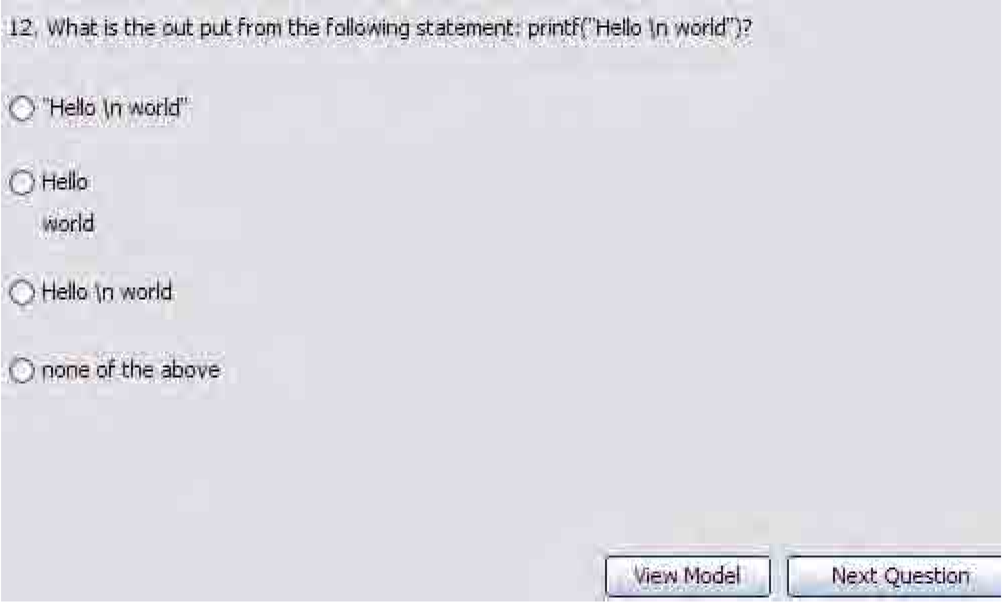
11. Which of the following should be used to display the string programming?

○ printf(programming);

○ printf("programming");

○ print("programming");

○ printf('programming');



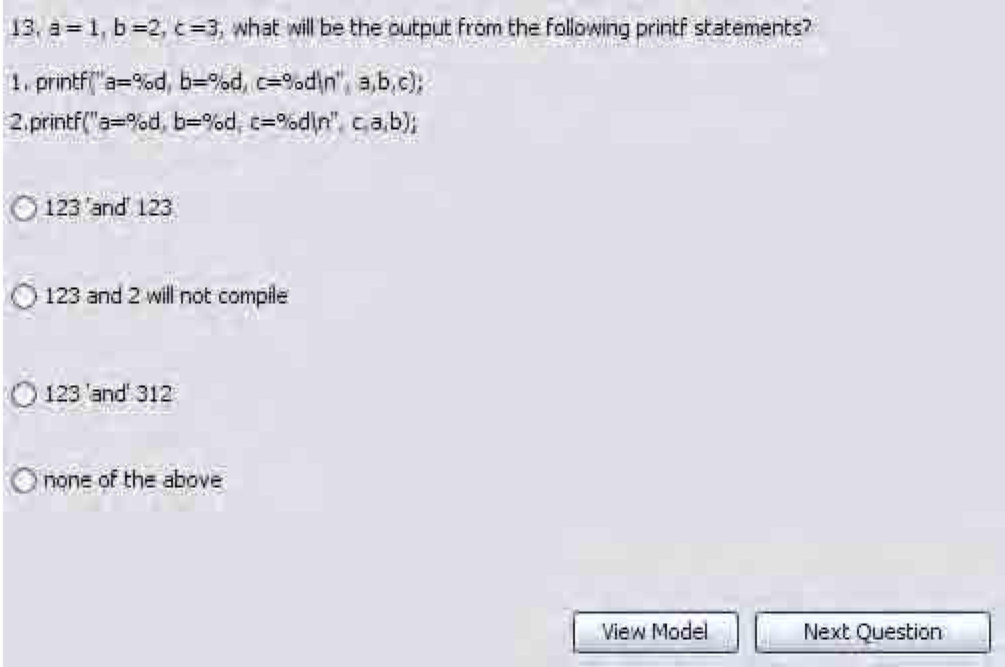
12. What is the out put from the following statement: printf("Hello \n world")?

○ "Hello \n world"

○ Hello
world

○ Hello \n world

○ none of the above



13. a = 1, b =2, c =3, what will be the output from the following printf statements?

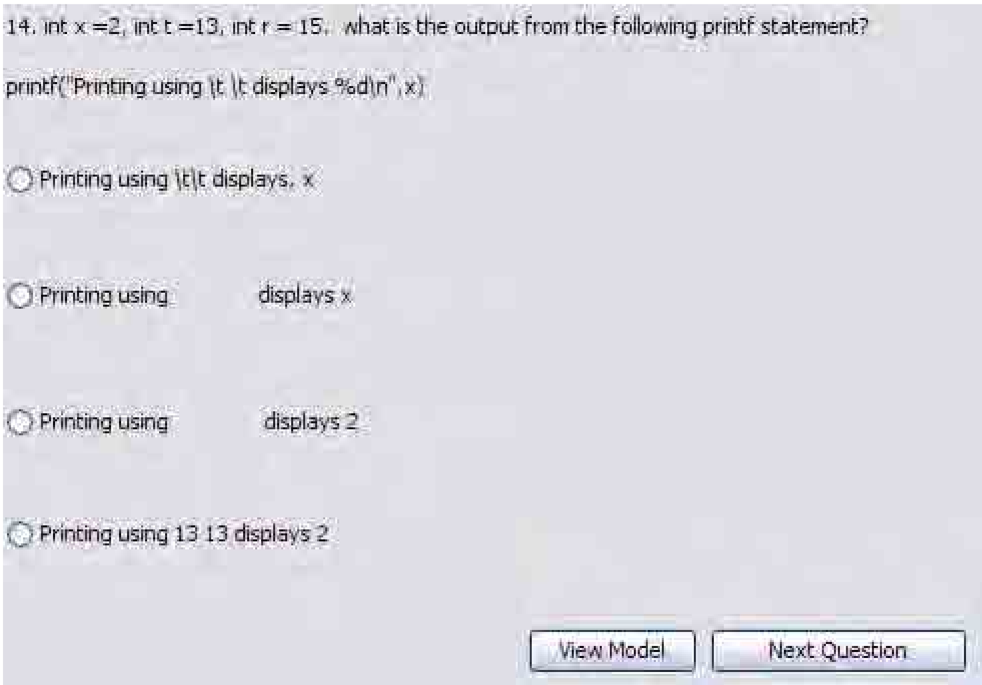
```
1. printf("a=%d, b=%d, c=%d\n", a,b,c);
2.printf("a=%d, b=%d, c=%d\n", c,a,b);
```

○ 123 'and' 123

○ 123 and 2 will not compile

○ 123 'and' 312

○ none of the above



14. int x =2, int t =13, int r = 15. what is the output from the following printf statement?

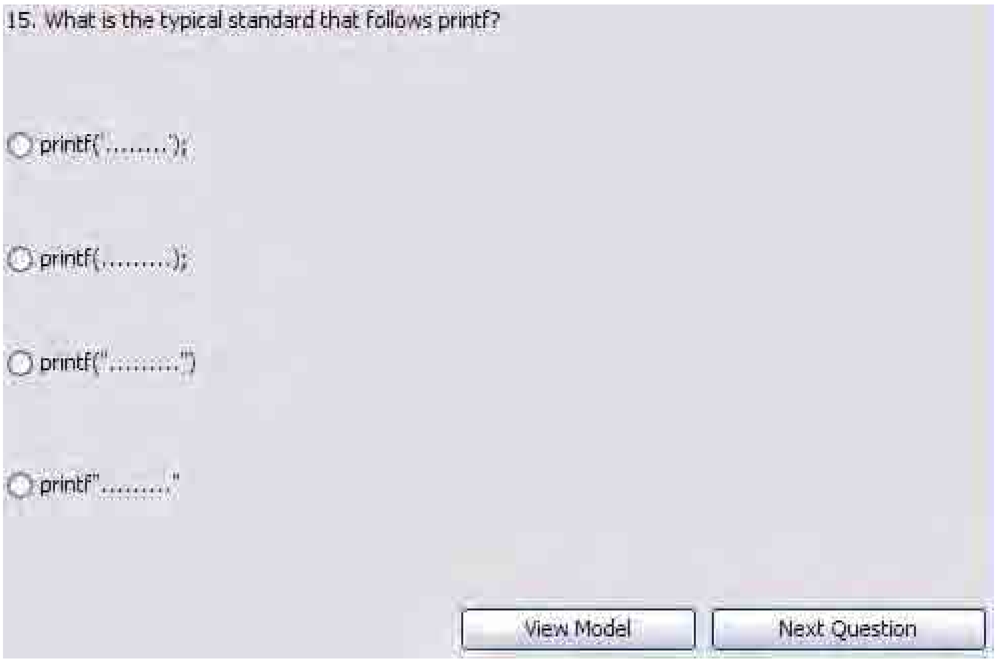
```
printf("Printing using \t \t displays %d\n",x)
```

○ Printing using \t\t displays. x

○ Printing using   displays x

○ Printing using   displays 2

○ Printing using 13 13 displays 2



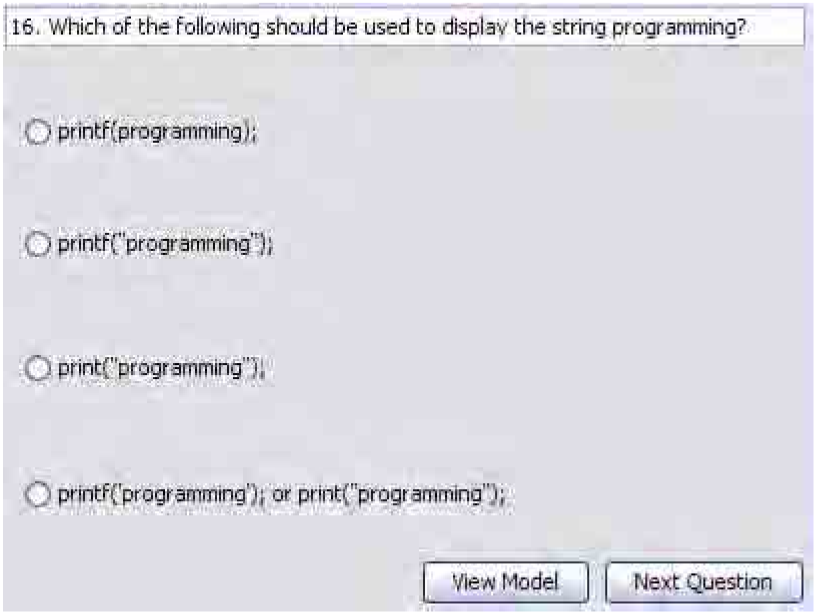
15. What is the typical standard that follows printf?

○ printf('........');

○ printf(.........);

○ printf(".........")

○ printf".........."



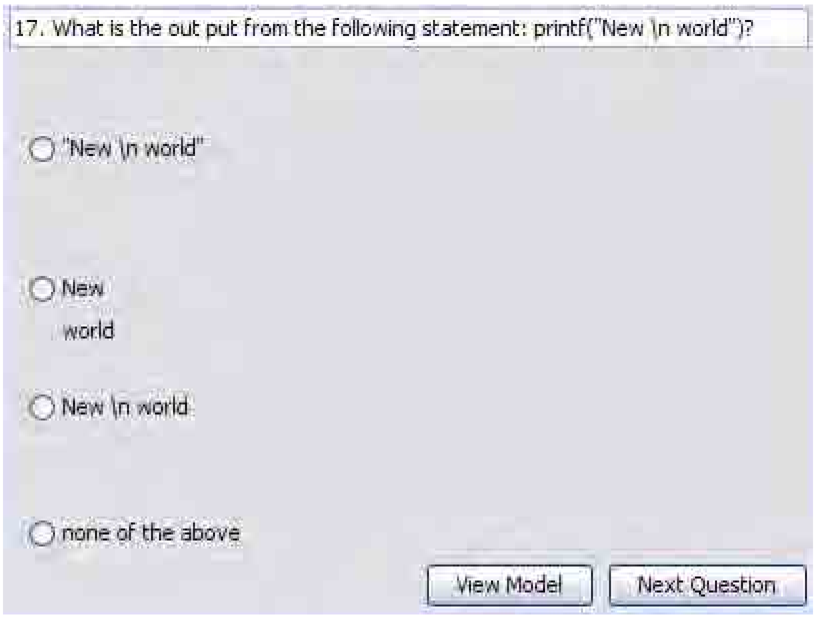
16. Which of the following should be used to display the string programming?

○ printf(programming);

○ printf("programming");

○ print("programming");

○ printf('programming'); or print("programming");



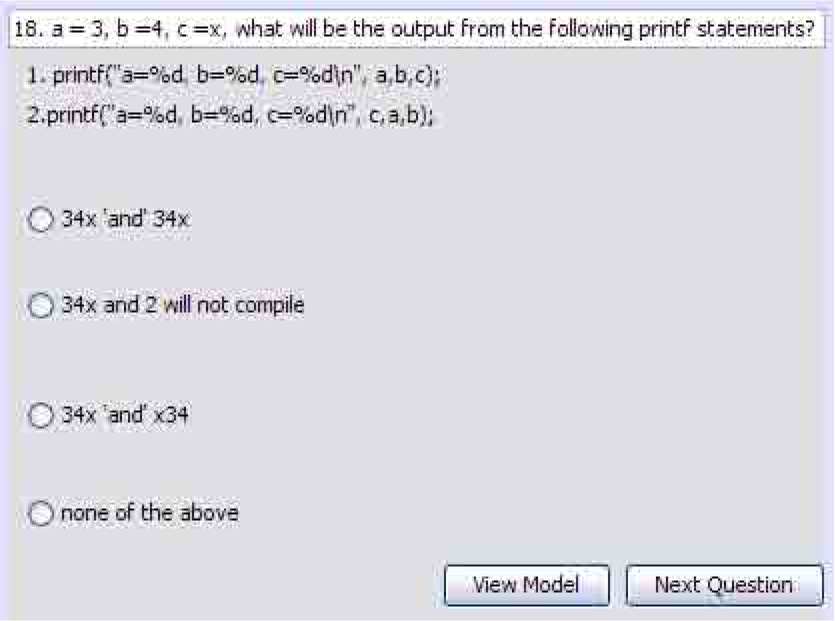
17. What is the out put from the following statement: printf("New \n world")?

○ "New \n world"

○ New
world

○ New \n world

○ none of the above



18. a = 3, b =4, c =x, what will be the output from the following printf statements?

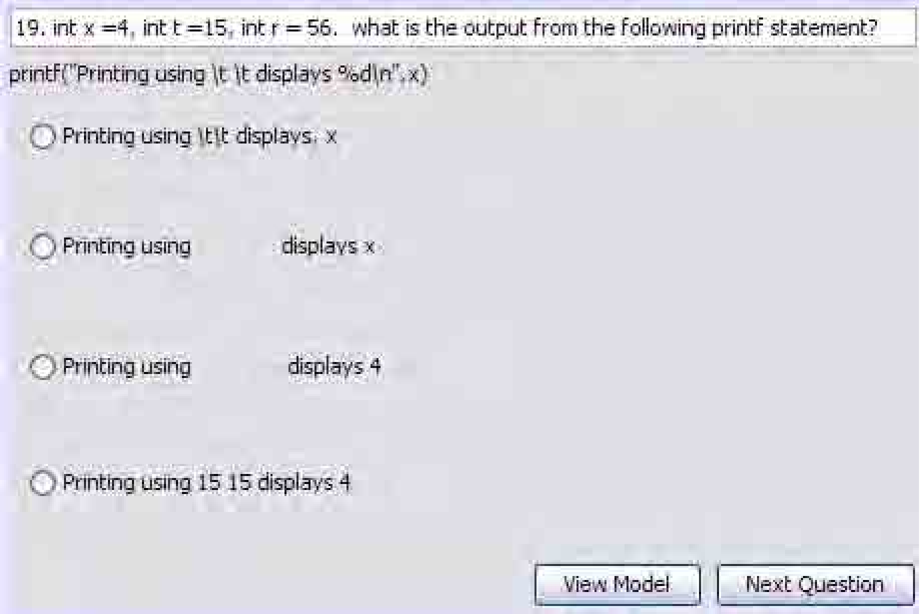
```
1. printf("a=%d, b=%d, c=%d\n", a,b,c);
2.printf("a=%d, b=%d, c=%d\n", c,a,b);
```

○ 34x 'and' 34x

○ 34x and 2 will not compile

○ 34x 'and' x34

○ none of the above



19. int x =4, int t =15, int r = 56. what is the output from the following printf statement?

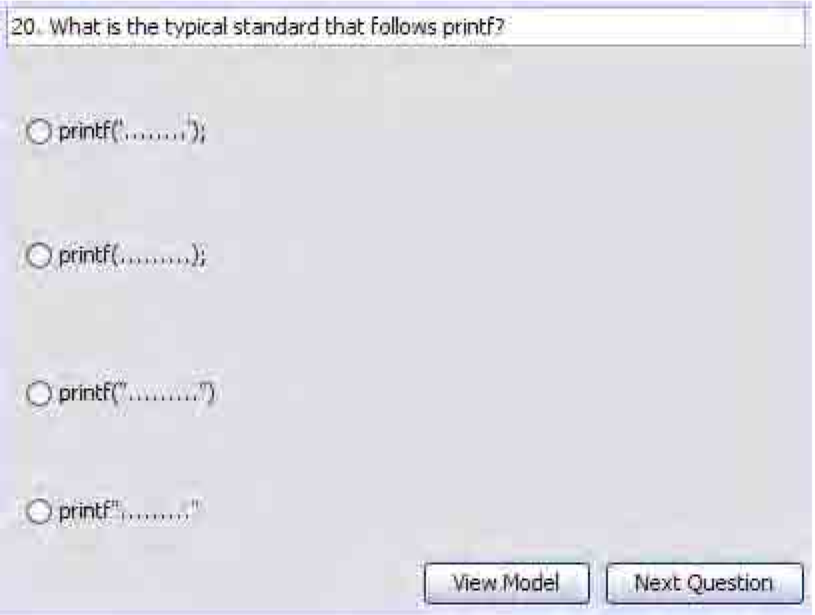
printf("Printing using \t \t displays %d\n", x)

○ Printing using \t\t displays. x

○ Printing using   displays x

○ Printing using   displays 4

○ Printing using 15 15 displays 4



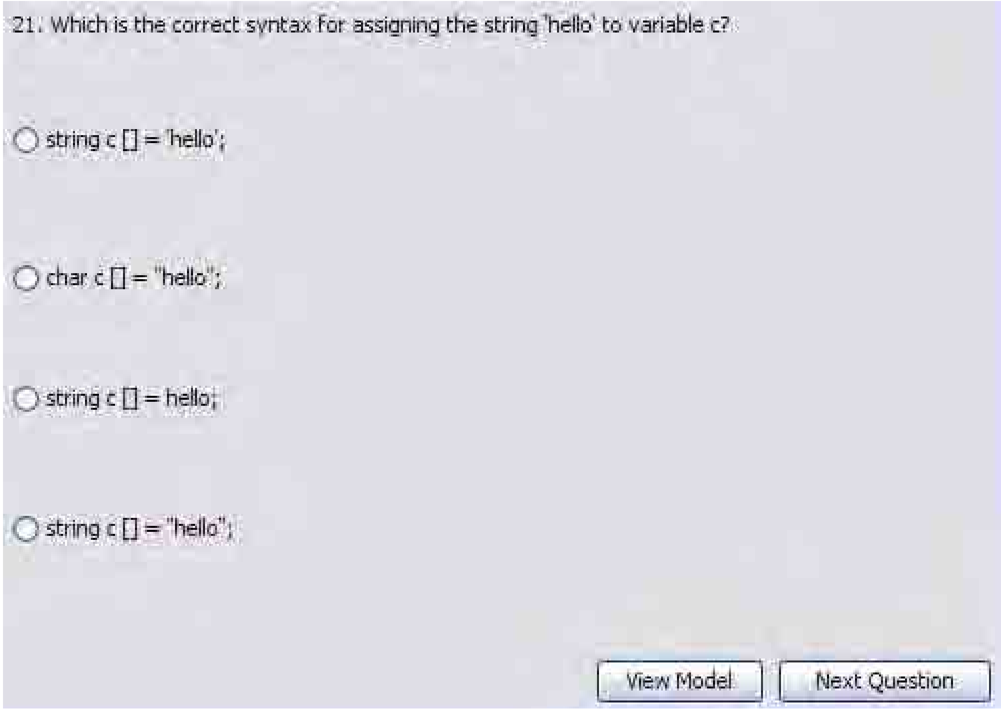
20. What is the typical standard that follows printf?

○ printf('........');

○ printf(.........);

○ printf(".........")

○ printf"........"



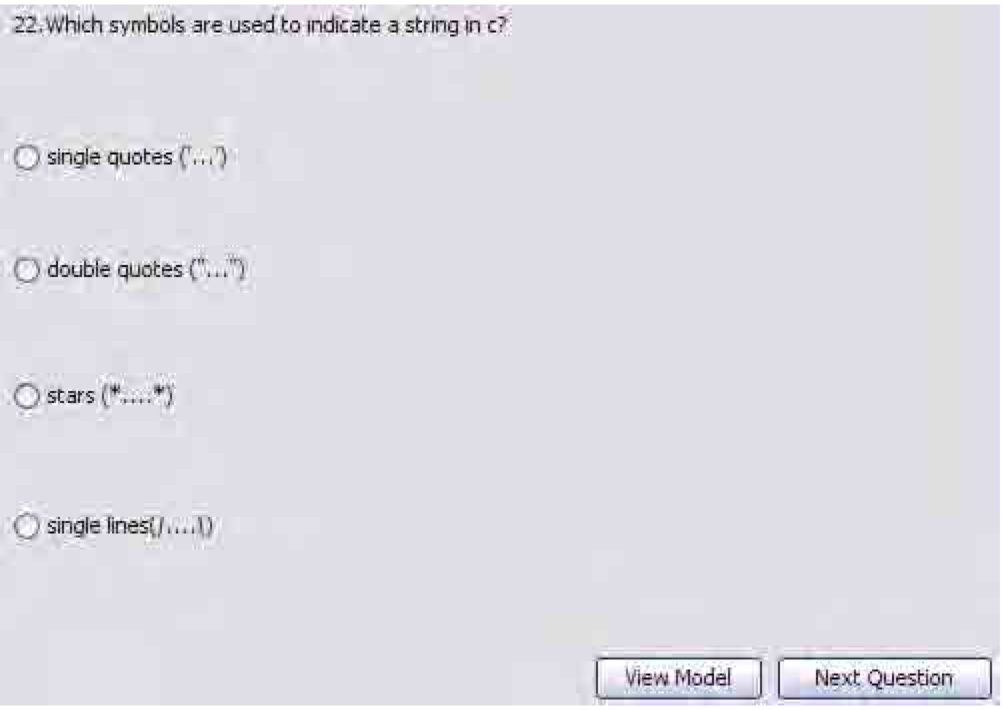
21. Which is the correct syntax for assigning the string 'hello' to variable c?

○ string c [] = 'hello';

○ char c [] = "hello";

○ string c [] = hello;

○ string c [] = "hello";



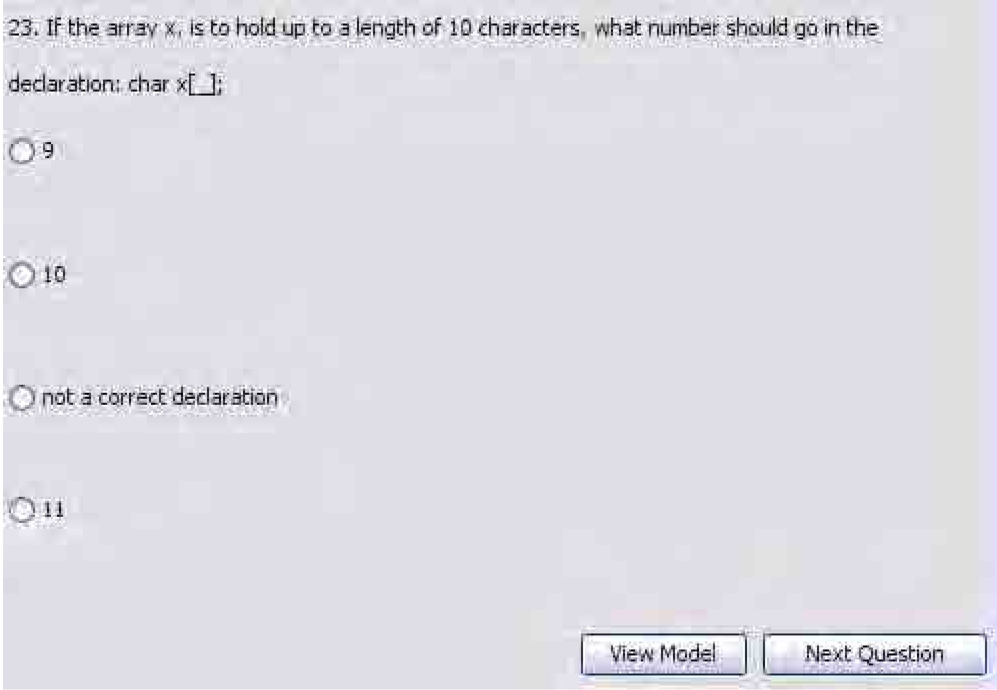
22. Which symbols are used to indicate a string in c?

○ single quotes ('...')

○ double quotes ("...")

○ stars (*....*)

○ single lines(/....\)



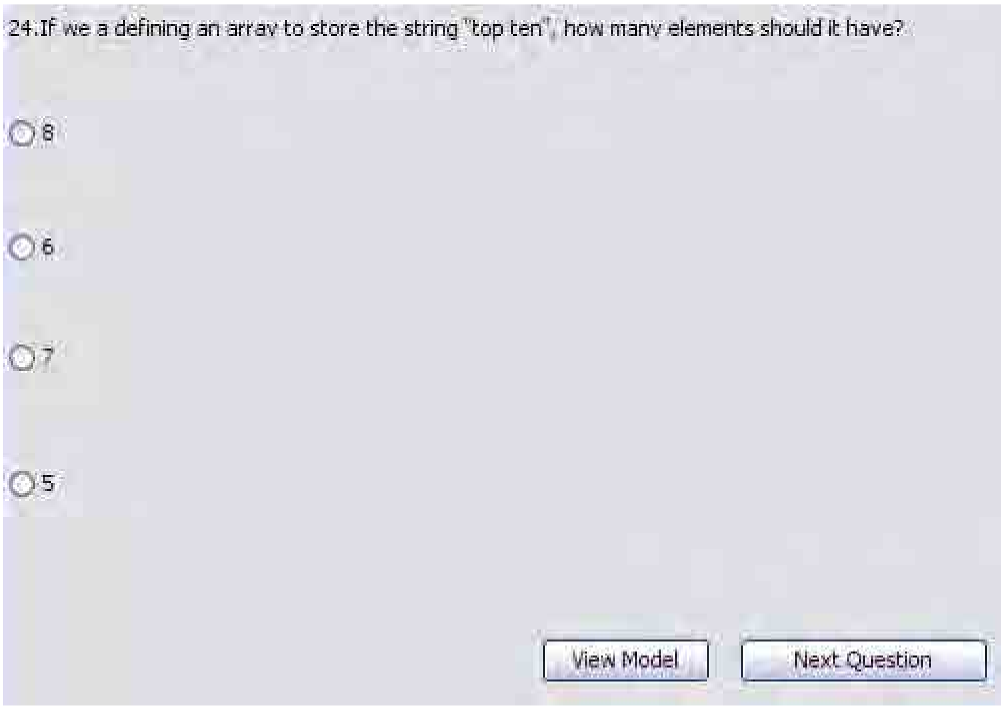
23. If the array x, is to hold up to a length of 10 characters, what number should go in the declaration: char x[_];

○ 9

○ 10

○ not a correct declaration

○ 11



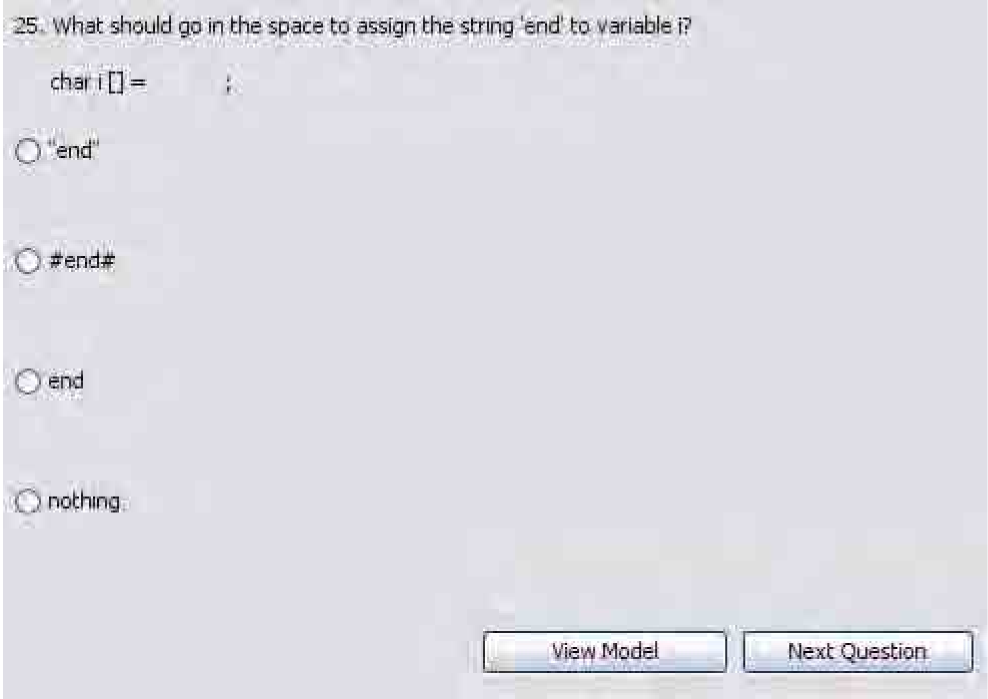
24.If we a defining an array to store the string "top ten", how many elements should it have?

○ 8

○ 6

○ 7

○ 5



25. What should go in the space to assign the string 'end' to variable i?

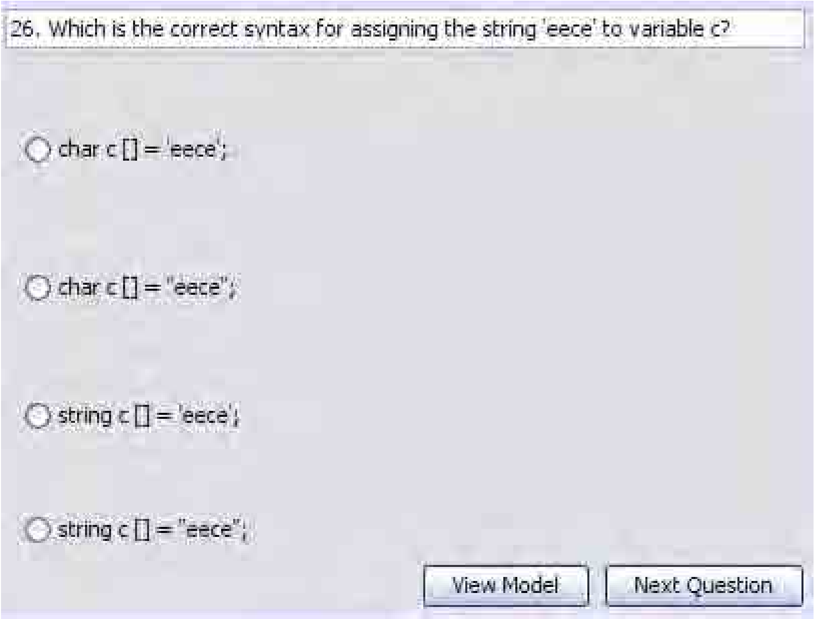
```
char i [] =          ;
```

○ "end"

○ #end#

○ end

○ nothing



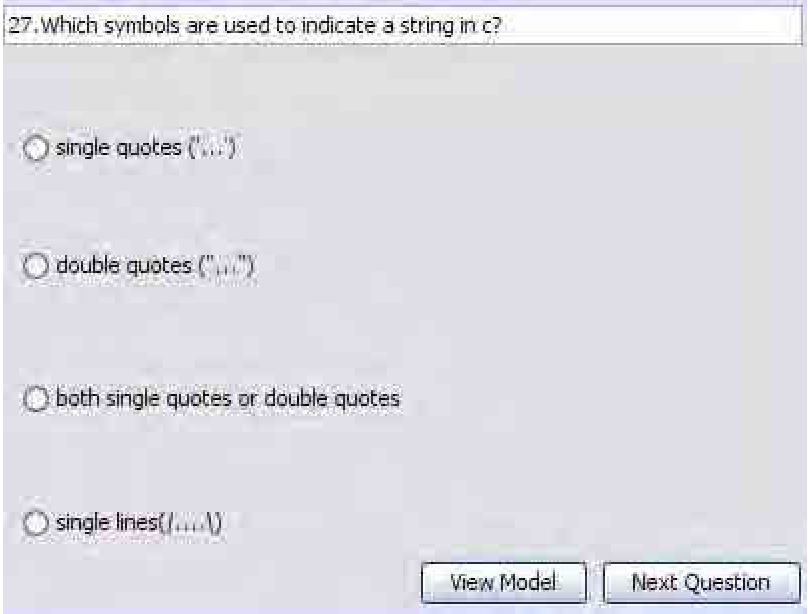
26. Which is the correct syntax for assigning the string 'eece' to variable c?

○ char c [] = 'eece';

○ char c [] = "eece";

○ string c [] = 'eece';

○ string c [] = "eece";



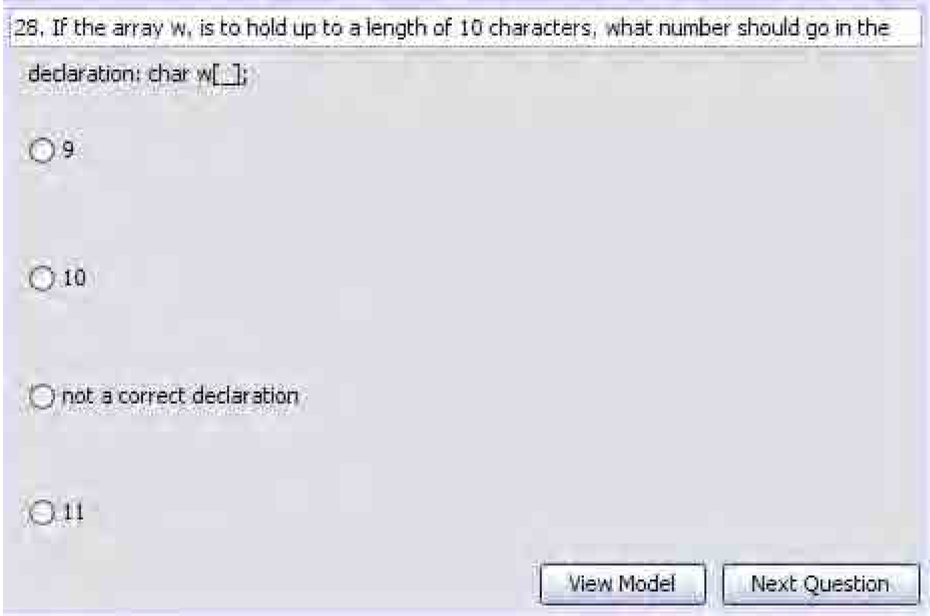
27.Which symbols are used to indicate a string in c?

○ single quotes ('...')

○ double quotes ("...")

○ both single quotes or double quotes

○ single lines(/....\)



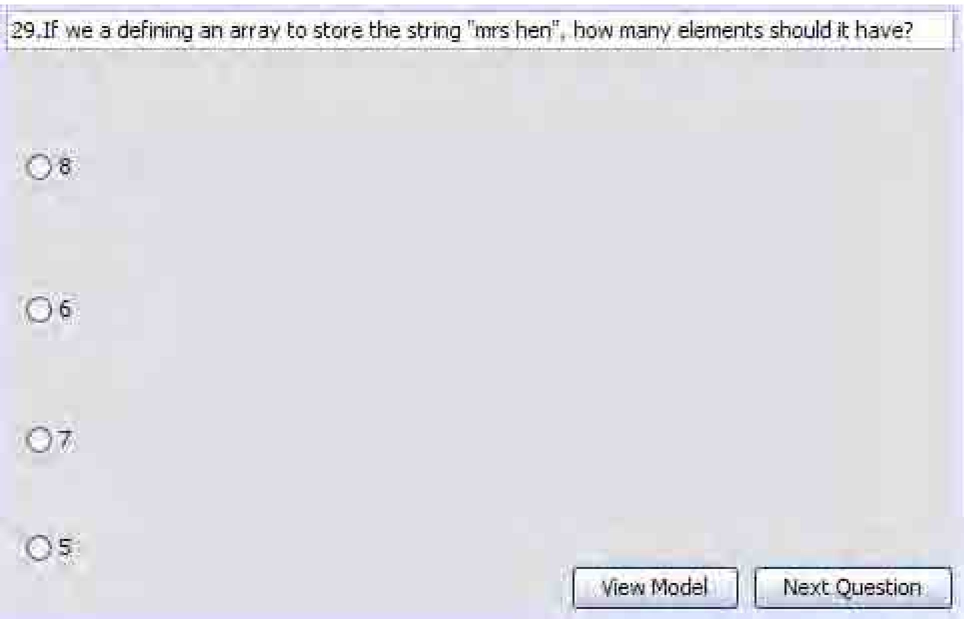
28. If the array w, is to hold up to a length of 10 characters, what number should go in the declaration: char w[_];

○ 9

○ 10

○ not a correct declaration

○ 11



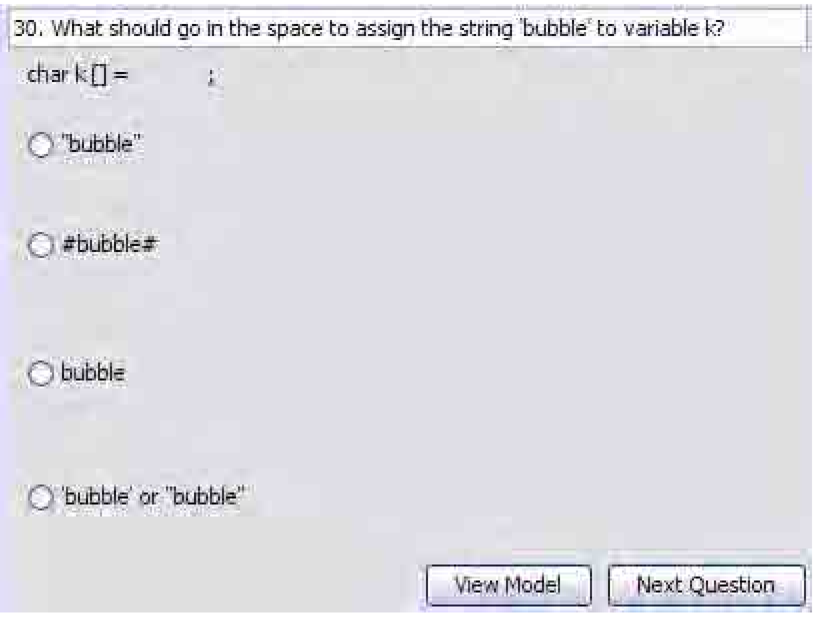
29.If we a defining an array to store the string "mrs hen", how many elements should it have?

○ 8

○ 6

○ 7

○ 5



30. What should go in the space to assign the string 'bubble' to variable k?

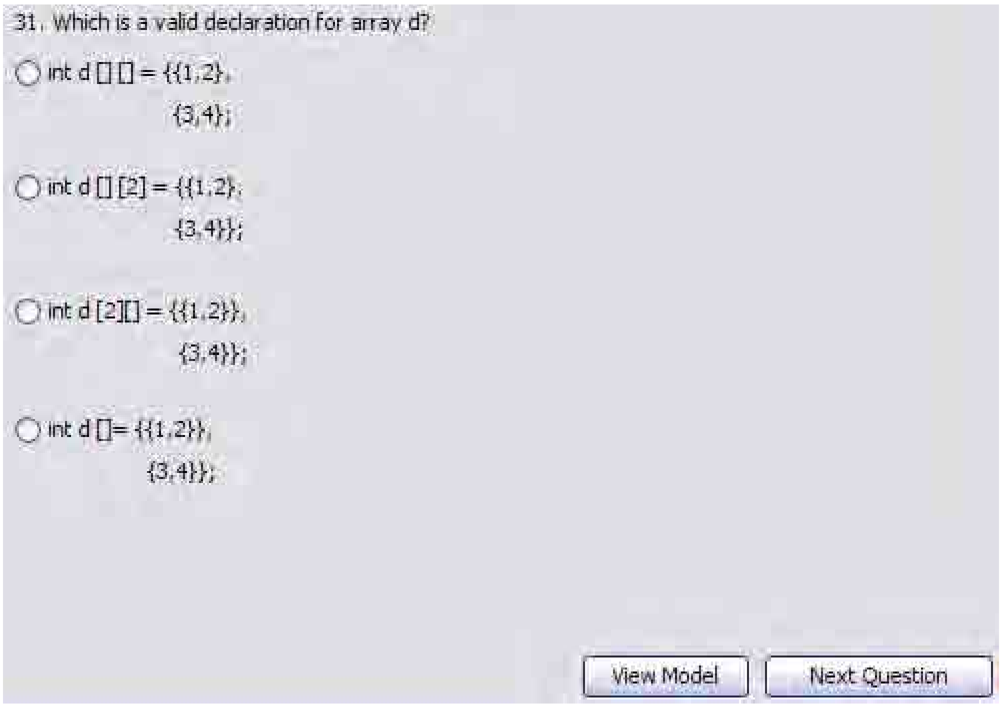
```
char k[] =          ;
```

○ "bubble"

○ #bubble#

○ bubble

○ 'bubble' or "bubble"



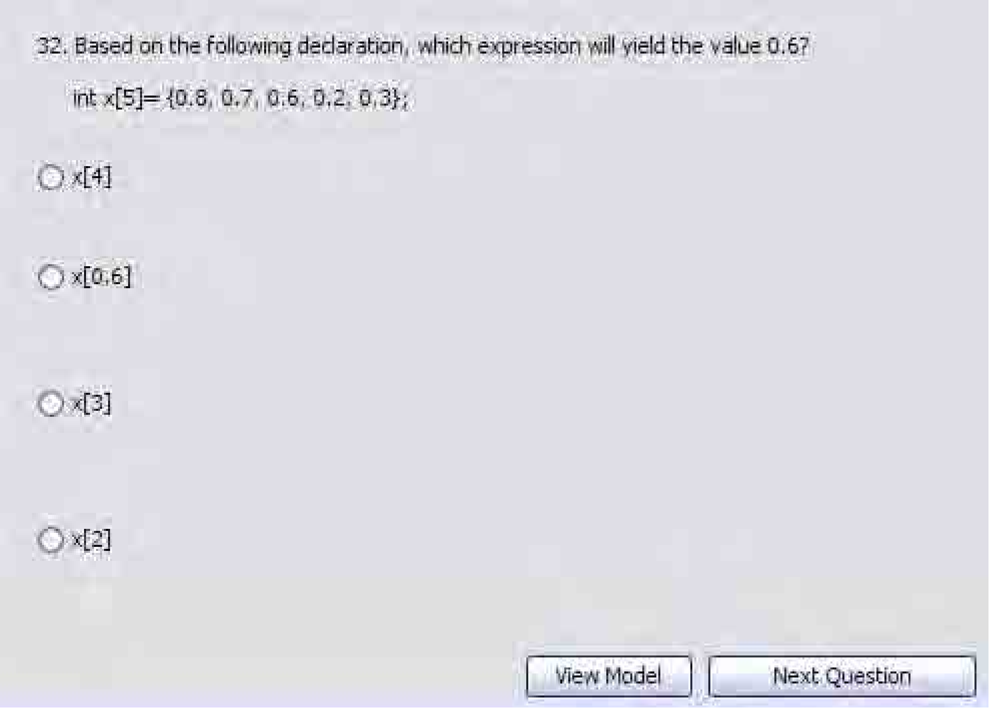
31. Which is a valid declaration for array d?

○ int d [] [] = {{1,2},
{3,4};

○ int d [] [2] = {{1,2},
{3,4}};

○ int d [2][] = {{1,2}},
{3,4}};

○ int d []= {{1,2}},
{3,4}};



32. Based on the following declaration, which expression will yield the value 0.6?

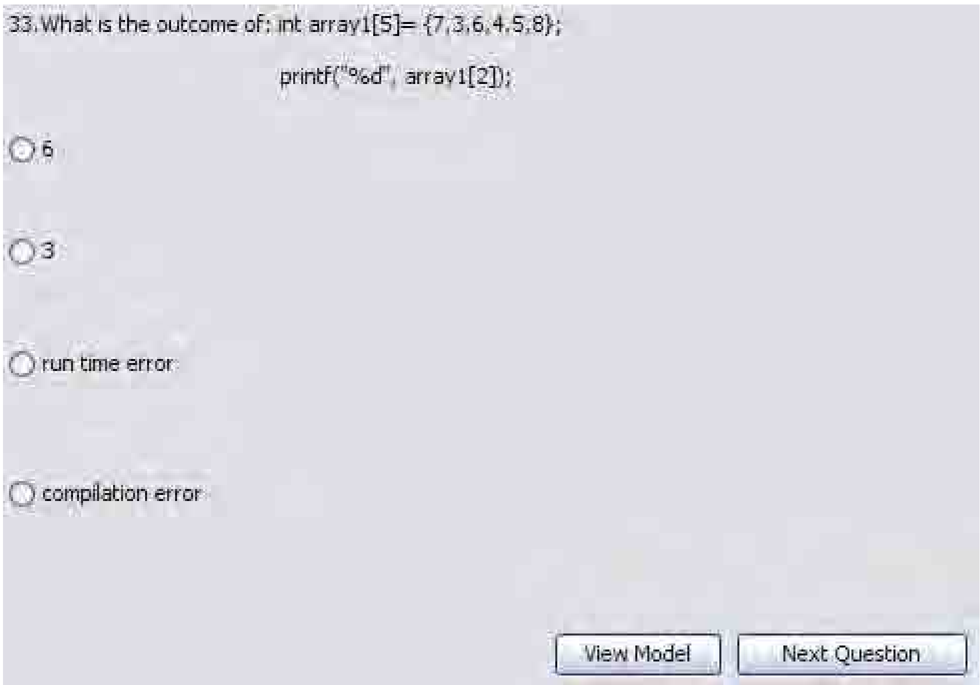
int x[5]= {0.8, 0.7, 0.6, 0.2, 0.3};

○ x[4]

○ x[0.6]

○ x[3]

○ x[2]



33.What is the outcome of: int array1[5]= {7,3,6,4,5,8};

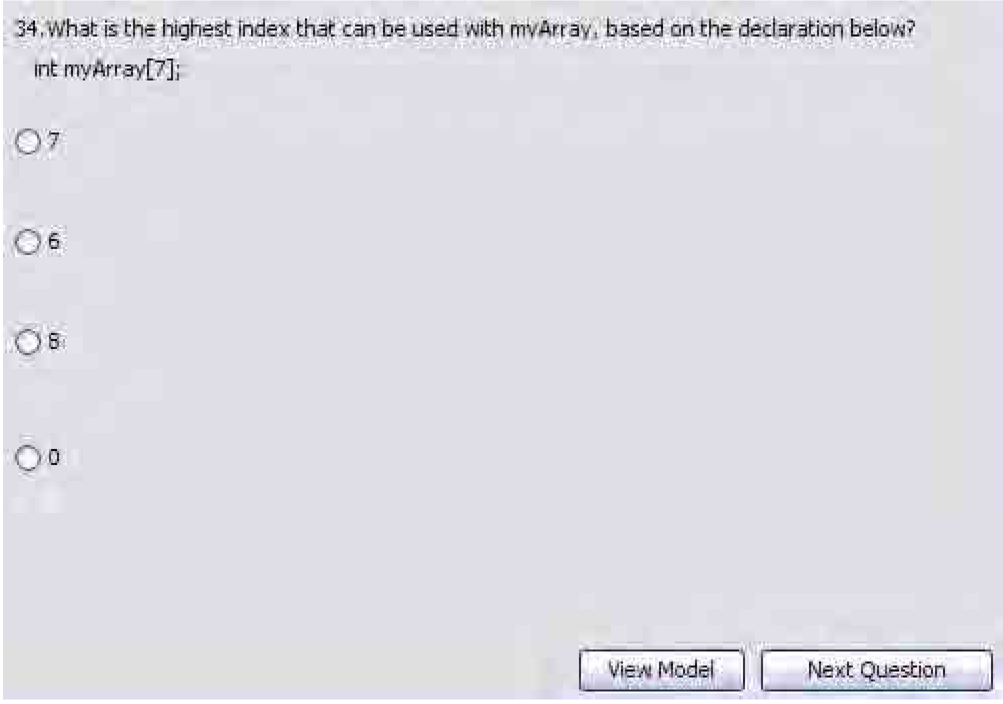
```
printf("%d", array1[2]);
```

○ 6

○ 3

○ run time error

○ compilation error



34.What is the highest index that can be used with myArray, based on the declaration below?

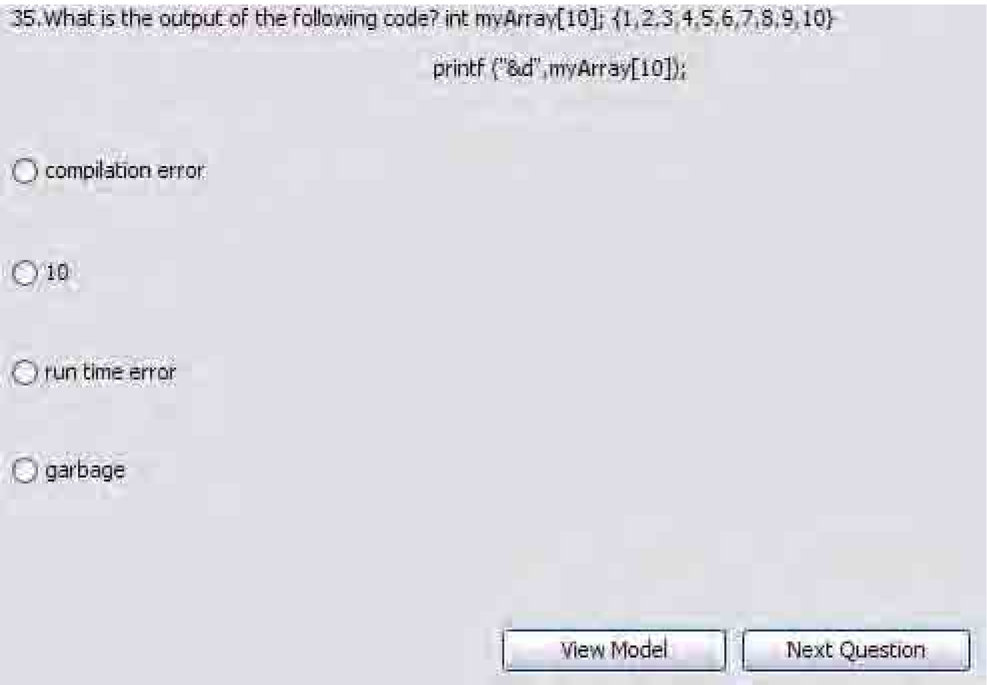
```
int myArray[7];
```

○ 7

○ 6

○ 8

○ 0



35.What is the output of the following code? int myArray[10]; {1,2,3,4,5,6,7,8,9,10}

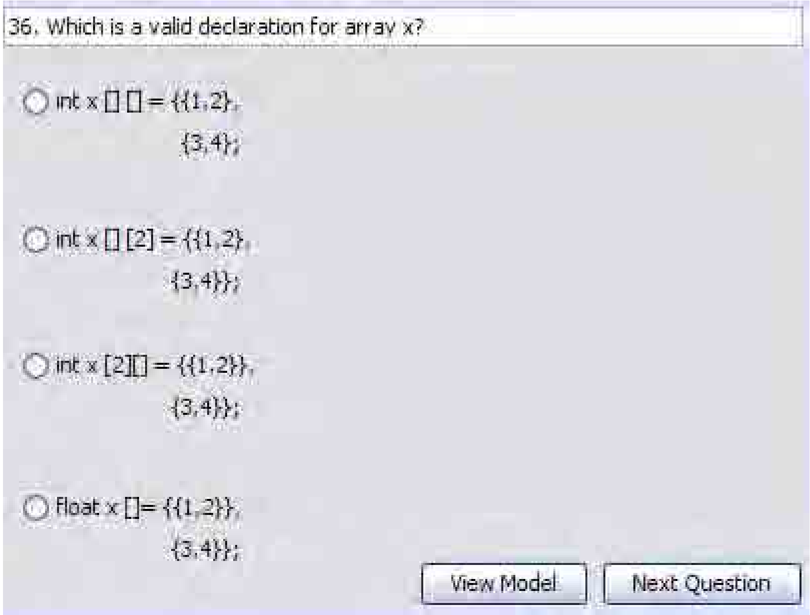
printf ("&d",myArray[10]);

○ compilation error

○ 10

○ run time error

○ garbage



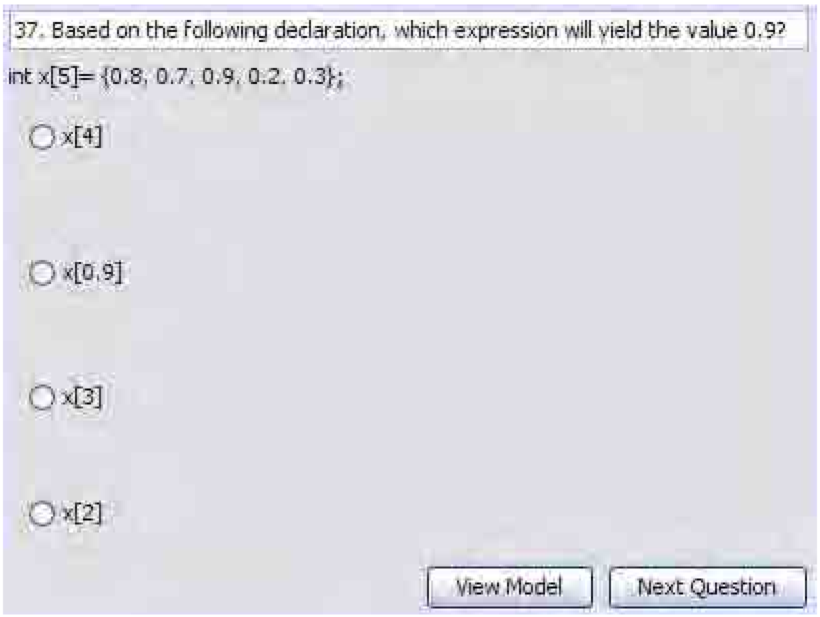
36. Which is a valid declaration for array x?

○ int x [] [] = {{1,2},
{3,4};

○ int x [] [2] = {{1,2},
{3,4}};

○ int x [2][] = {{1,2}},
{3,4}};

○ float x []= {{1,2}},
{3,4}};



37. Based on the following declaration, which expression will yield the value 0.9?

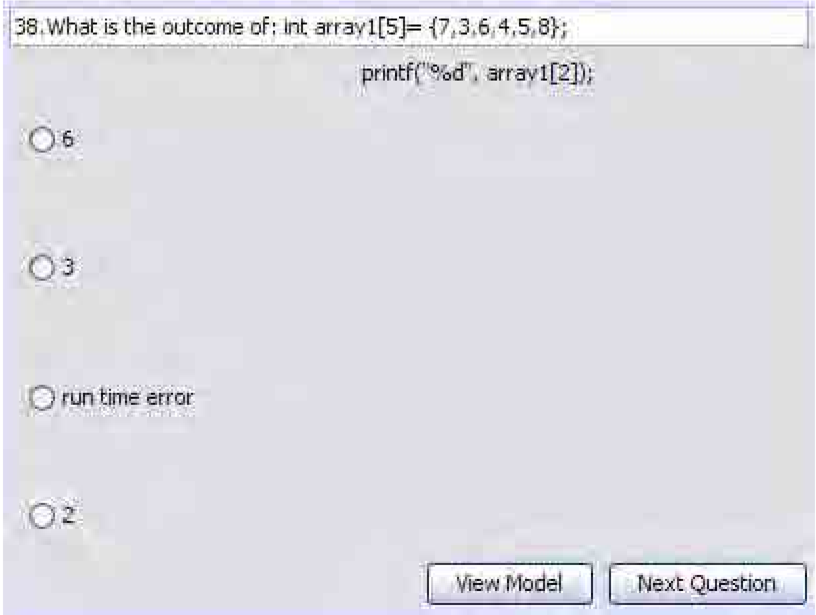
int x[5]= {0.8, 0.7, 0.9, 0.2, 0.3};

○ x[4]

○ x[0.9]

○ x[3]

○ x[2]



38.What is the outcome of: int array1[5]= {7,3,6,4,5,8};

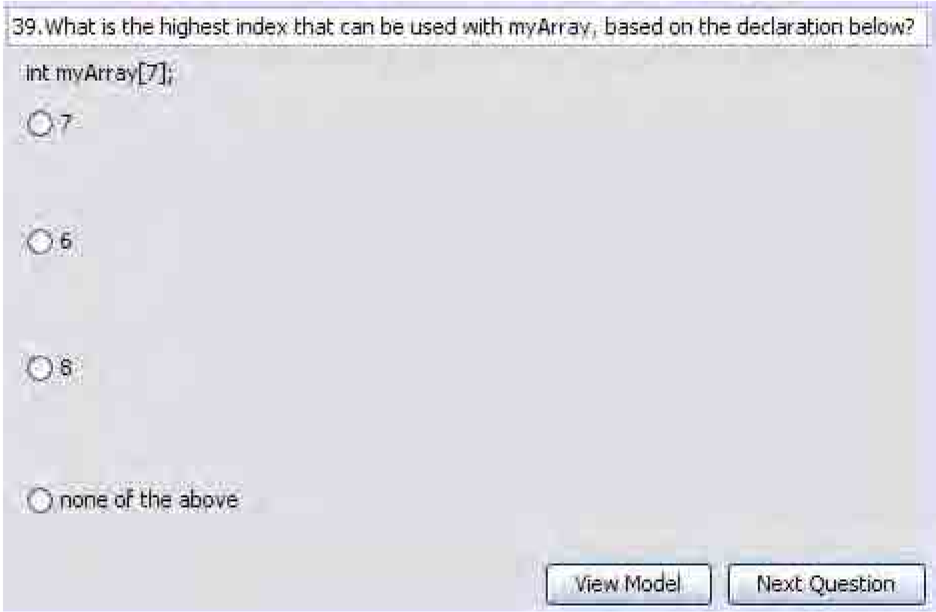
printf("%d", array1[2]);

○ 6

○ 3

○ run time error

○ 2



39.What is the highest index that can be used with myArray, based on the declaration below?

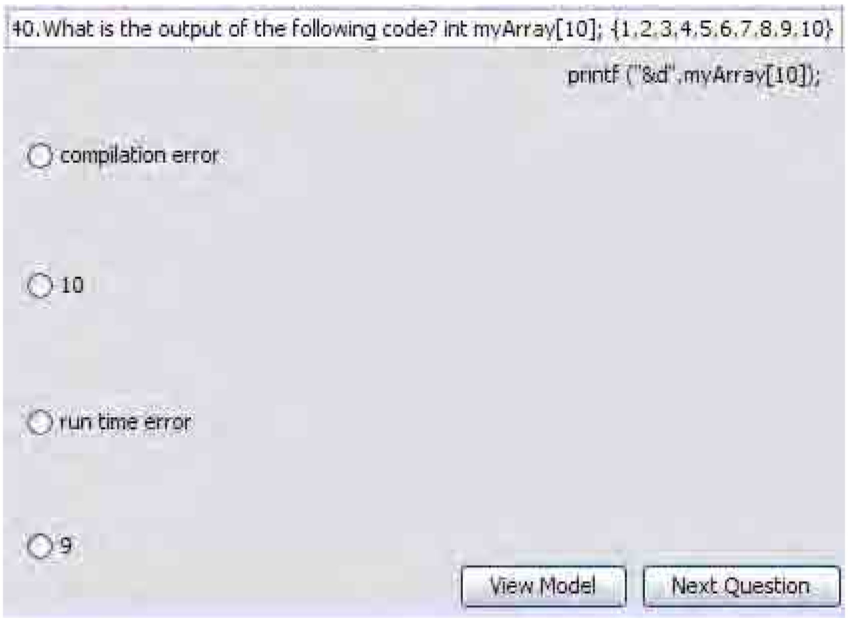
```
int myArray[7];
```

○ 7

○ 6

○ 8

○ none of the above



40.What is the output of the following code? int myArray[10]; {1,2,3,4,5,6,7,8,9,10}

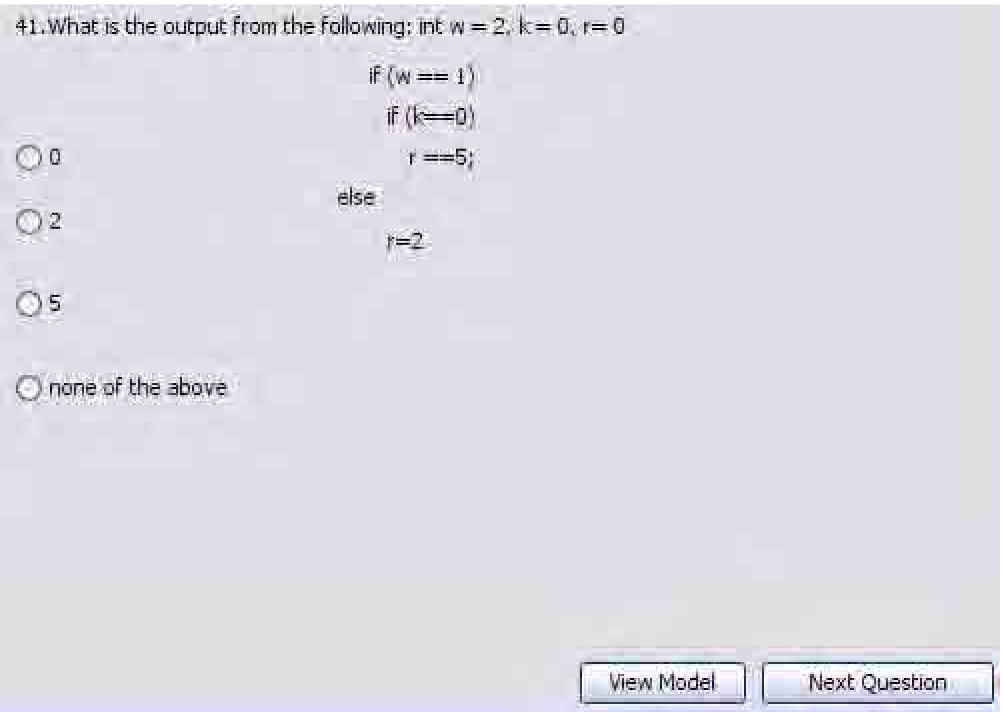
```
printf ("&d",myArray[10]);
```

○ compilation error

○ 10

○ run time error

○ 9



41.What is the output from the following: int w = 2, k = 0, r= 0

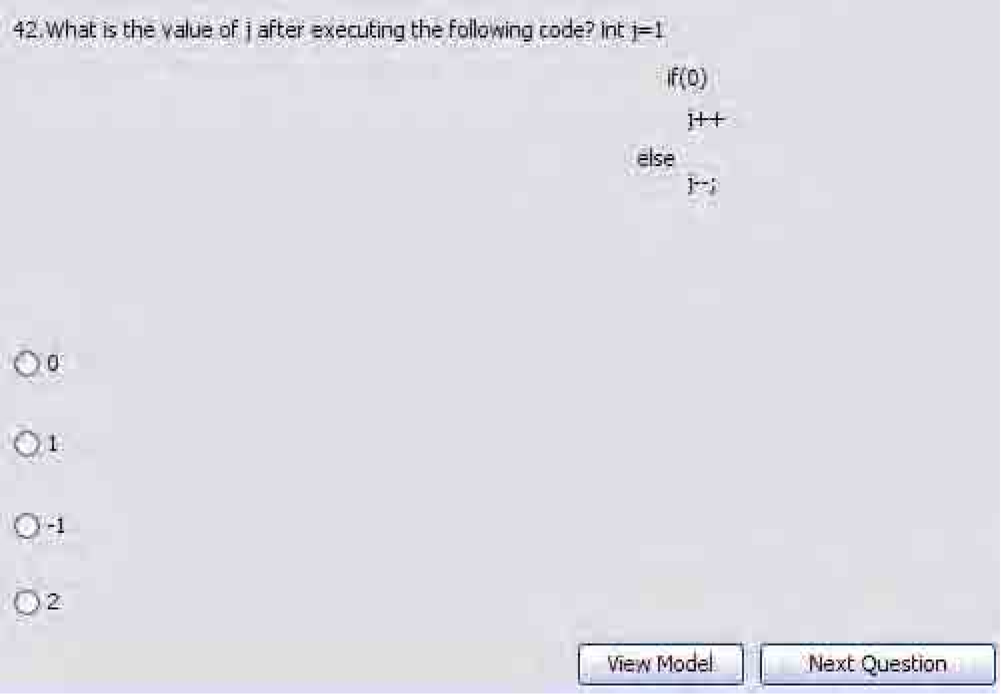
```
        if (w == 1)
          if (k==0)
             r ==5;
      else
          r=2
```

○ 0

○ 2

○ 5

○ none of the above



42.What is the value of j after executing the following code? int j=1

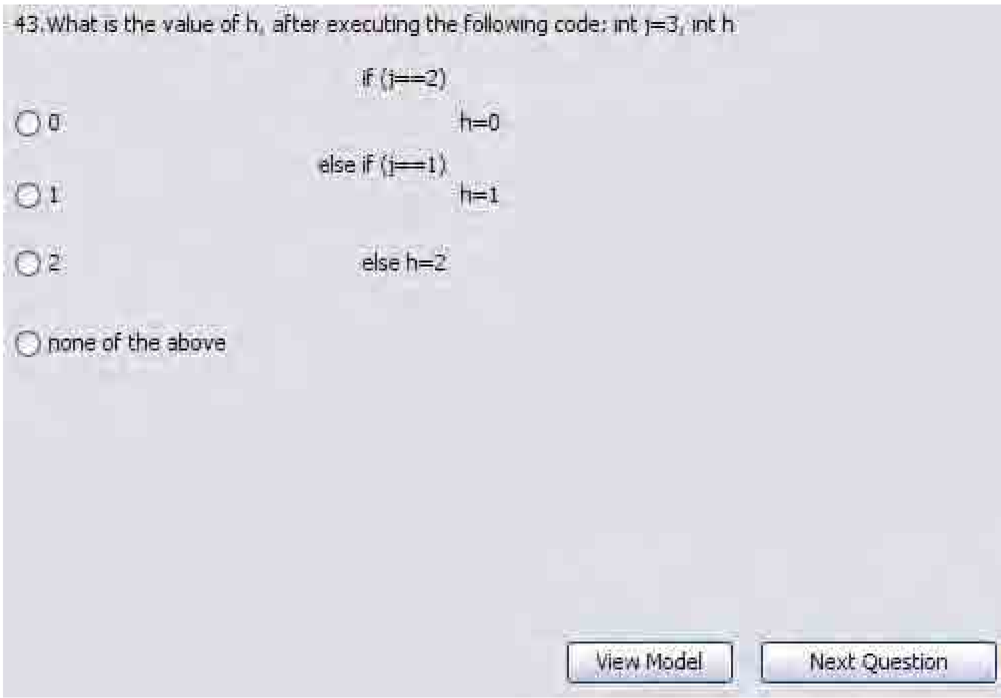
```
  if(0)
     j++
else
     j--;
```

○ 0

○ 1

○ -1

○ 2



43.What is the value of h, after executing the following code: int j=3, int h

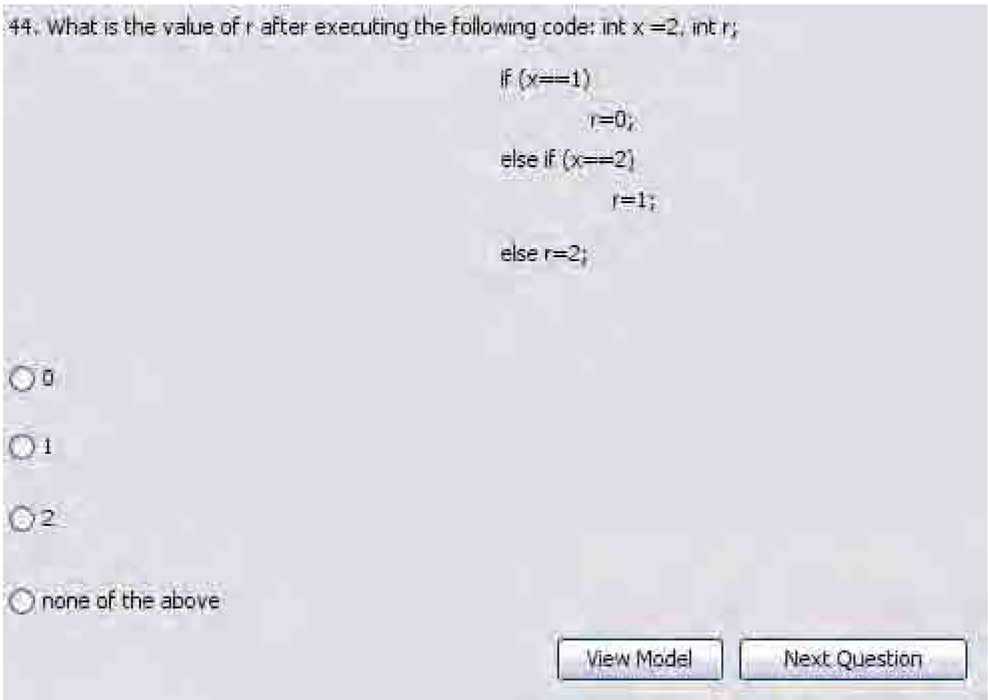
```
if (j==2)
        h=0
else if (j==1)
        h=1
else h=2
```

○ 0

○ 1

○ 2

○ none of the above



44. What is the value of r after executing the following code: int x =2, int r;

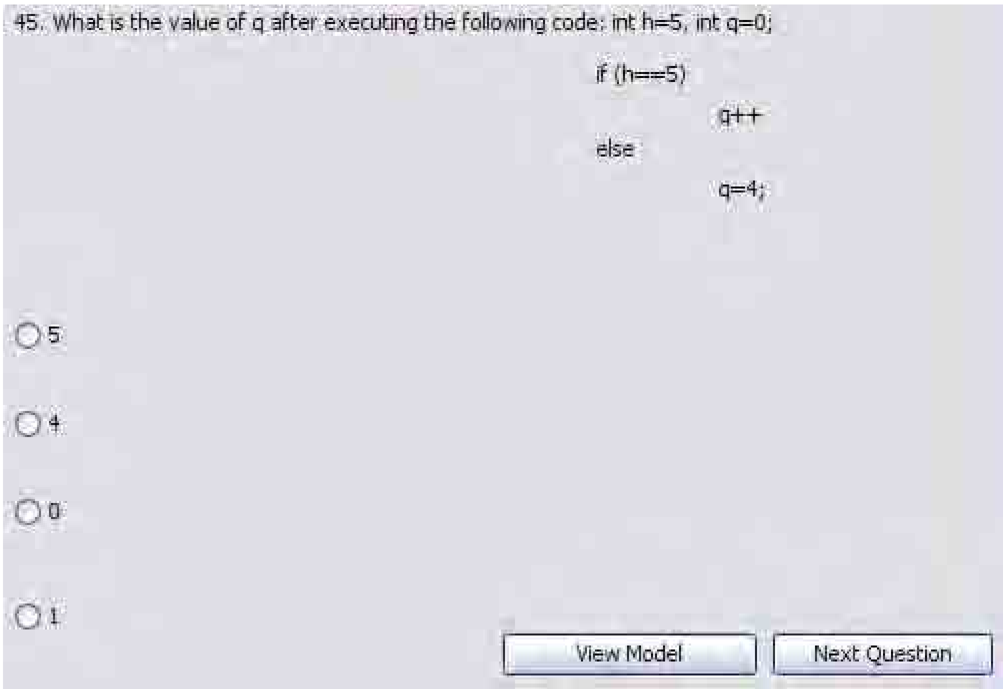
```
if (x==1)
        r=0;
else if (x==2)
        r=1;
else r=2;
```

○ 0

○ 1

○ 2

○ none of the above



45. What is the value of q after executing the following code: int h=5, int q=0;

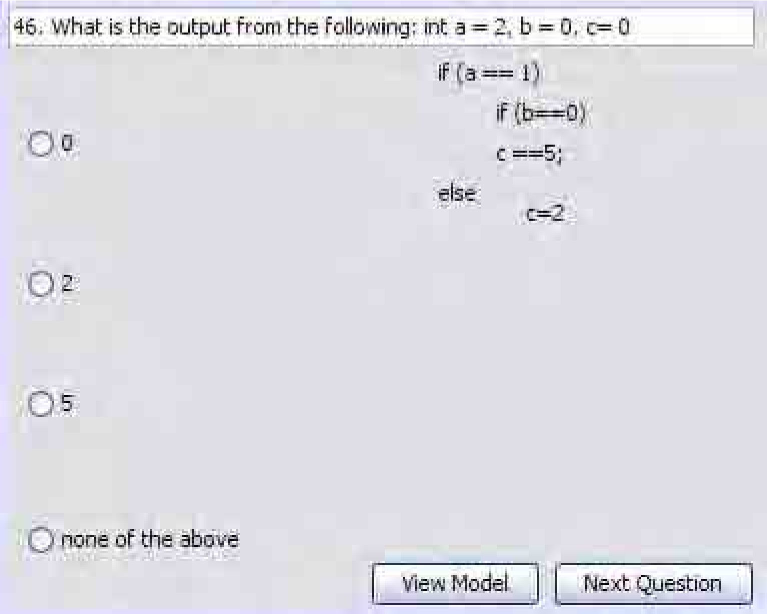
```
if (h==5)
          q++
else
          q=4;
```

○ 5

○ 4

○ 0

○ 1



46. What is the output from the following: int a = 2, b = 0, c= 0

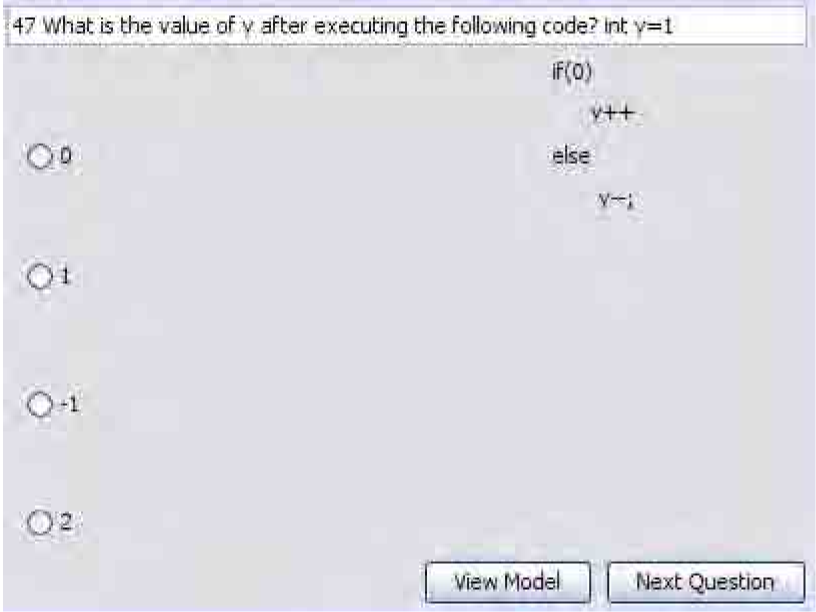
```
if (a == 1)
     if (b==0)
     c ==5;
else
       c=2
```

○ 0

○ 2

○ 5

○ none of the above



47 What is the value of y after executing the following code? int y=1

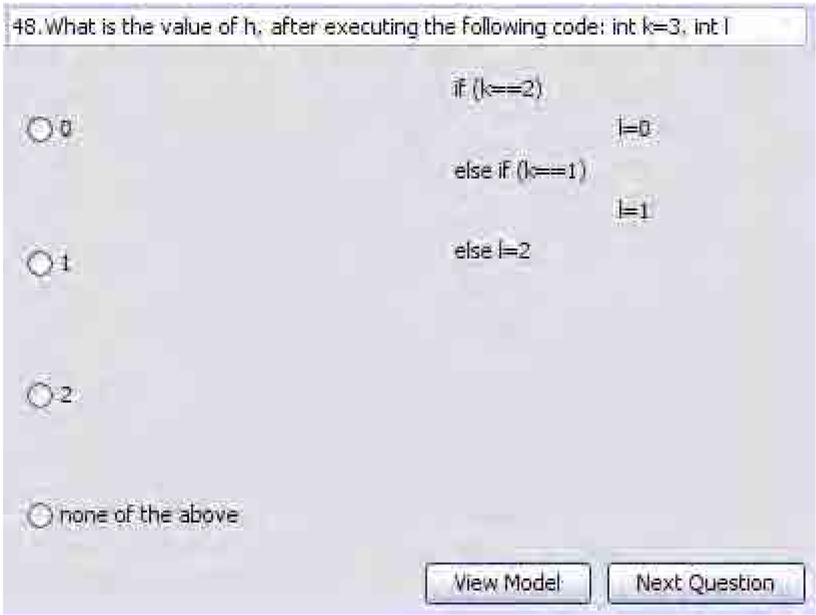
```
if(0)
    y++
else
    y--;
```

○ 0

○ 1

○ -1

○ 2



48.What is the value of h, after executing the following code: int k=3, int l

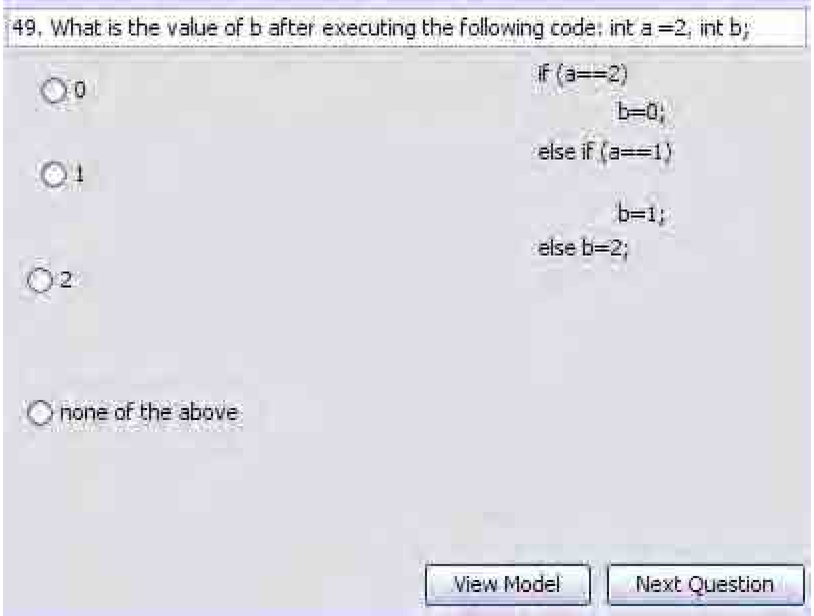
```
if (k==2)
            l=0
else if (k==1)
            l=1
else l=2
```

○ 0

○ 1

○ 2

○ none of the above



49. What is the value of b after executing the following code: int a =2, int b;

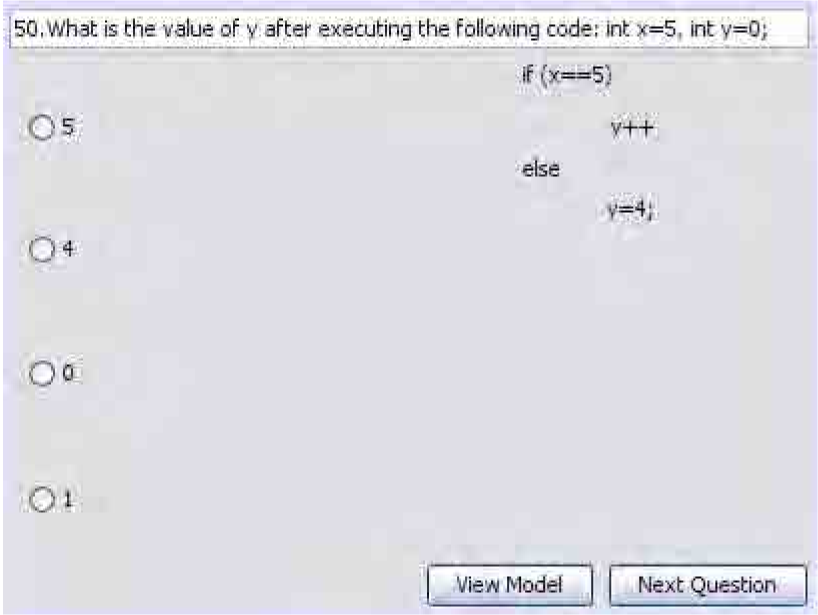
```
if (a==2)
        b=0;
else if (a==1)

        b=1;
else b=2;
```

○ 0

○ 1

○ 2

○ none of the above



50.What is the value of y after executing the following code: int x=5, int y=0;

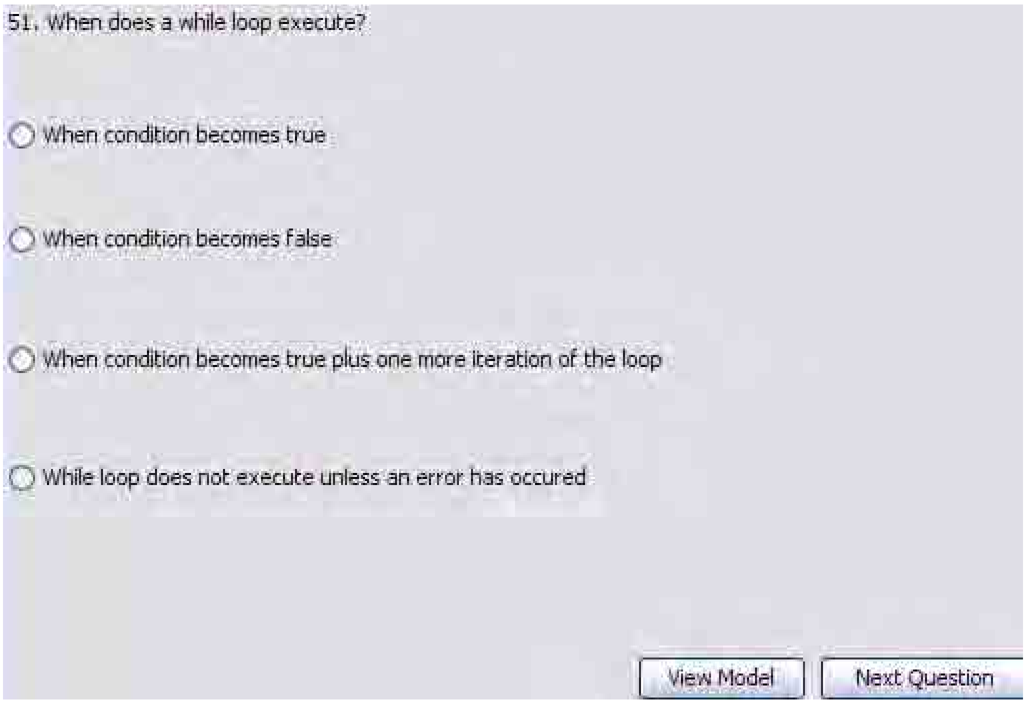
```
if (x==5)
        y++
else
        y=4;
```

○ 5

○ 4

○ 0

○ 1



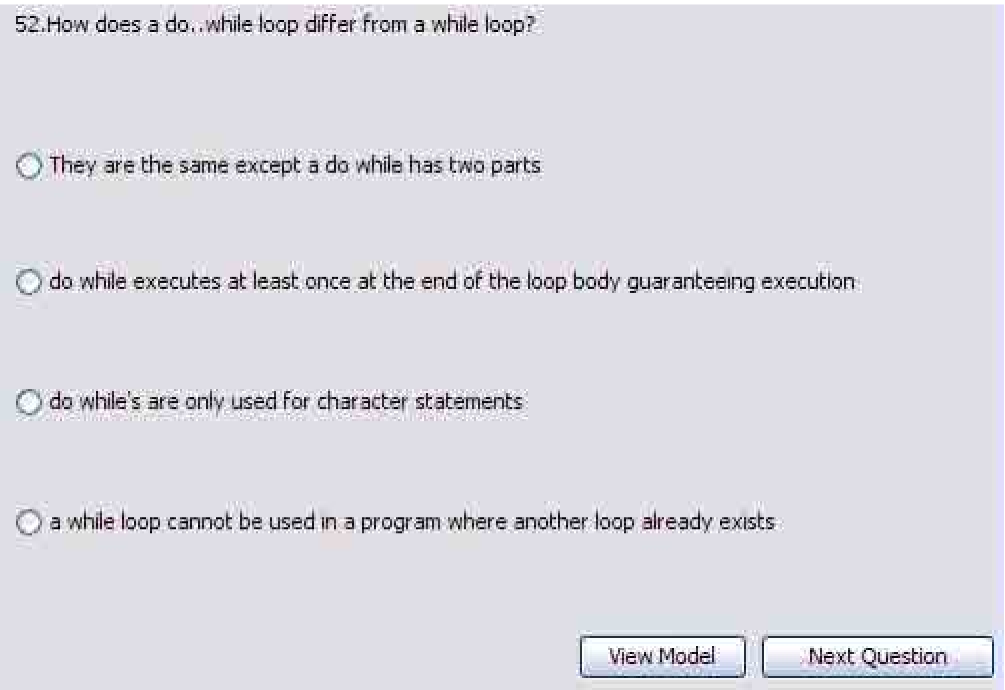
51. When does a while loop execute?

○ When condition becomes true

○ When condition becomes false

○ When condition becomes true plus one more iteration of the loop

○ While loop does not execute unless an error has occured



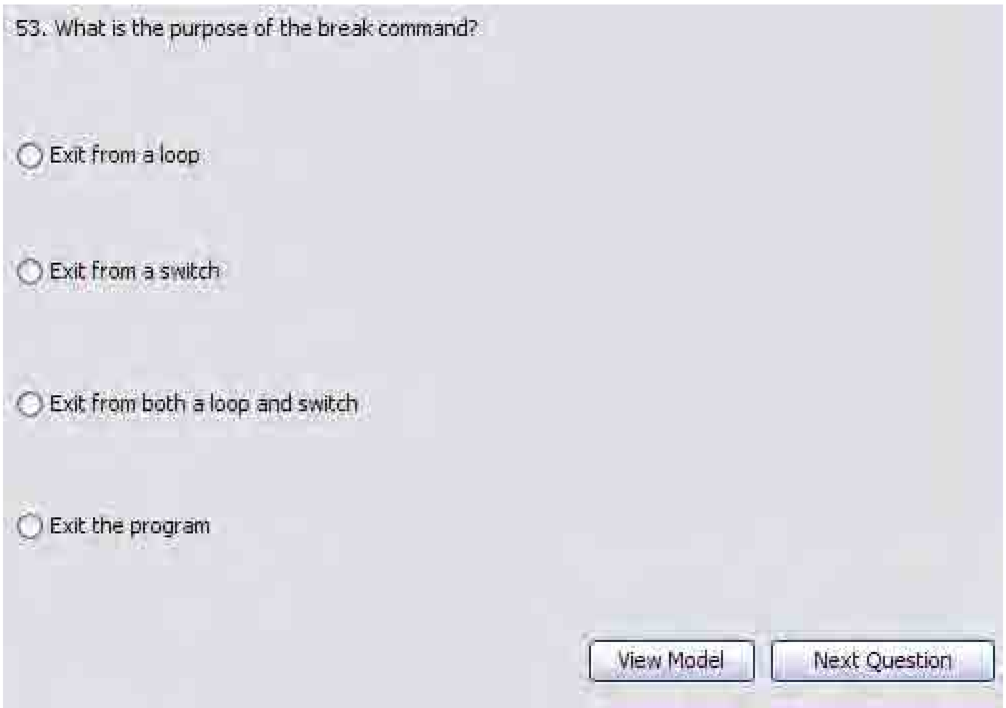
52.How does a do..while loop differ from a while loop?

○ They are the same except a do while has two parts

○ do while executes at least once at the end of the loop body guaranteeing execution

○ do while's are only used for character statements

○ a while loop cannot be used in a program where another loop already exists



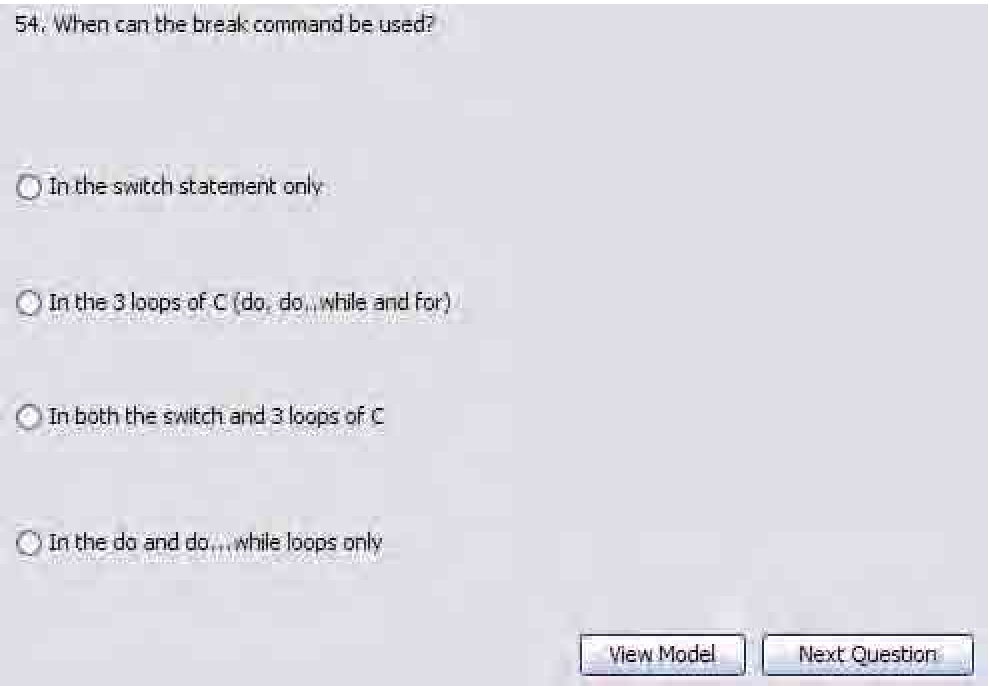
53. What is the purpose of the break command?

○ Exit from a loop

○ Exit from a switch

○ Exit from both a loop and switch

○ Exit the program



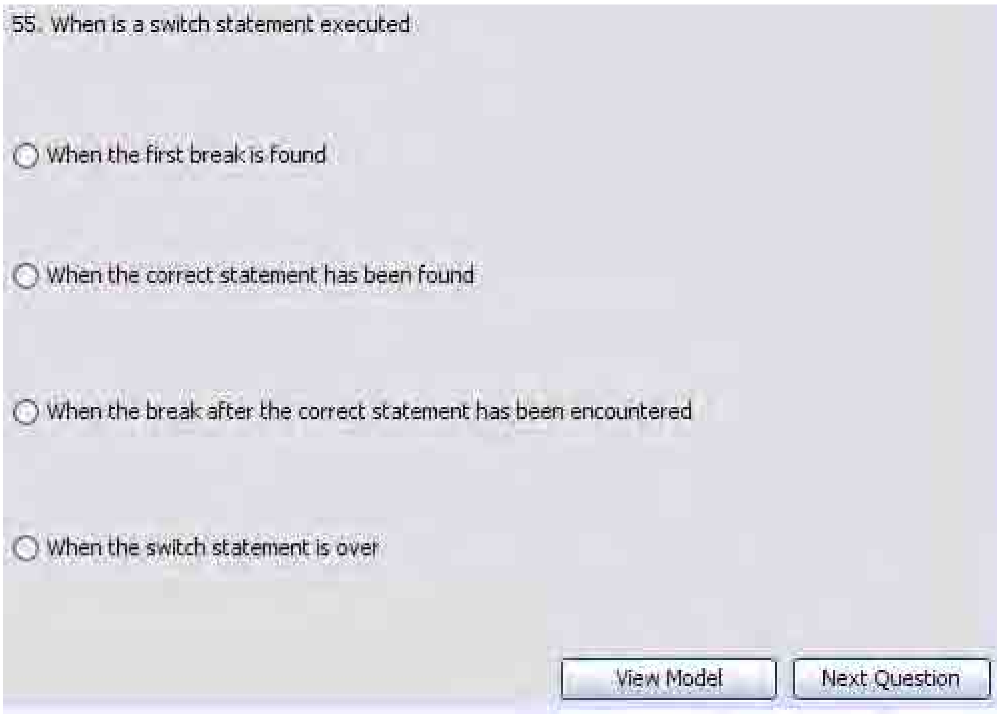
54. When can the break command be used?

○ In the switch statement only

○ In the 3 loops of C (do, do...while and for)

○ In both the switch and 3 loops of C

○ In the do and do....while loops only



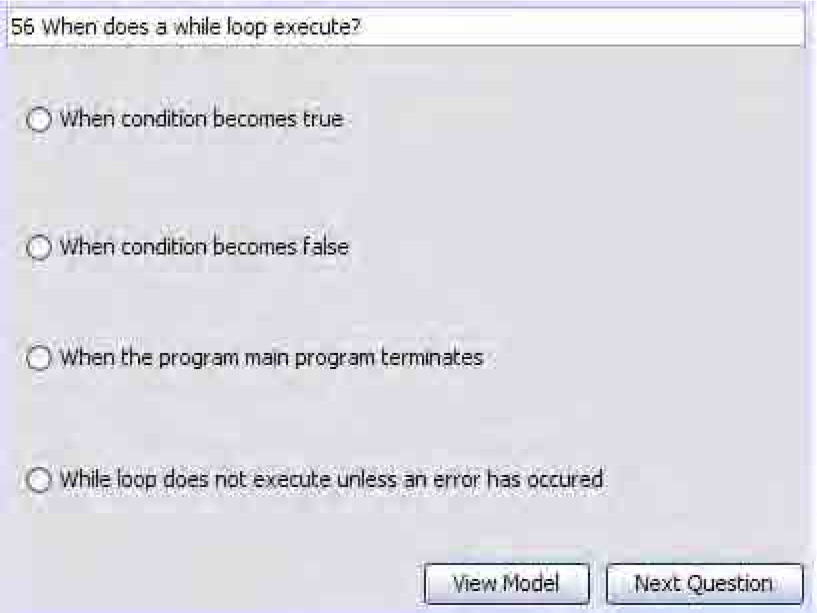
55. When is a switch statement executed

○ When the first break is found

○ When the correct statement has been found

○ When the break after the correct statement has been encountered

○ When the switch statement is over



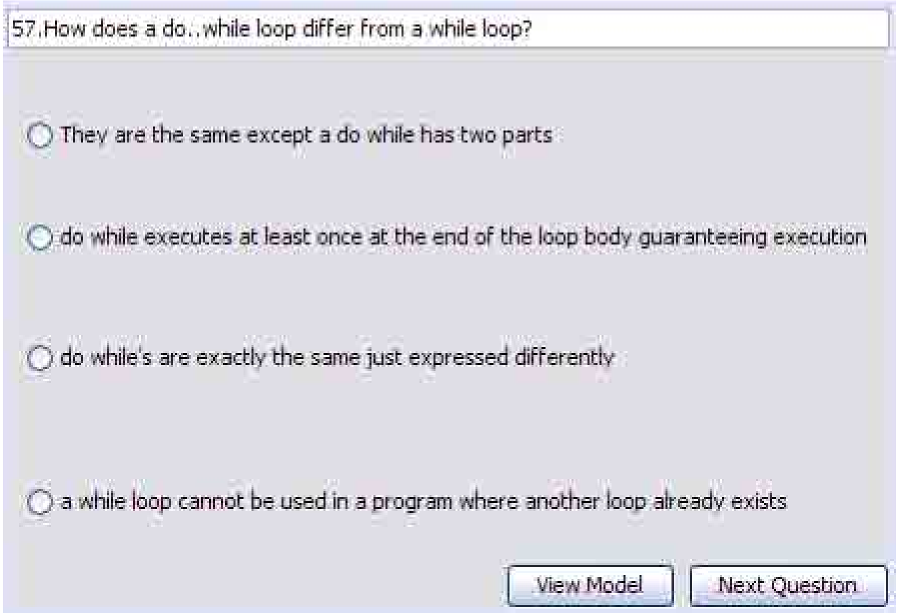
56 When does a while loop execute?

○ When condition becomes true

○ When condition becomes false

○ When the program main program terminates

○ While loop does not execute unless an error has occured



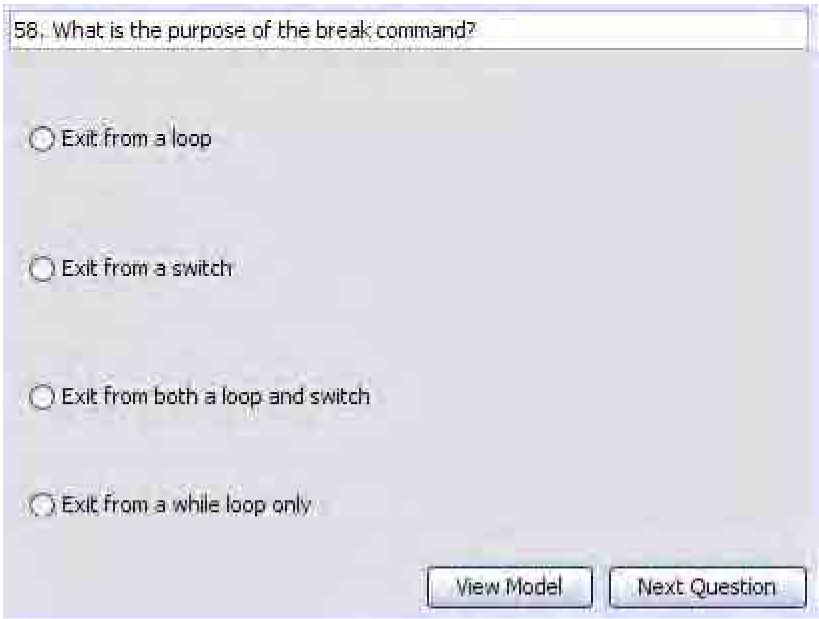
57.How does a do..while loop differ from a while loop?

○ They are the same except a do while has two parts

○ do while executes at least once at the end of the loop body guaranteeing execution

○ do while's are exactly the same just expressed differently

○ a while loop cannot be used in a program where another loop already exists



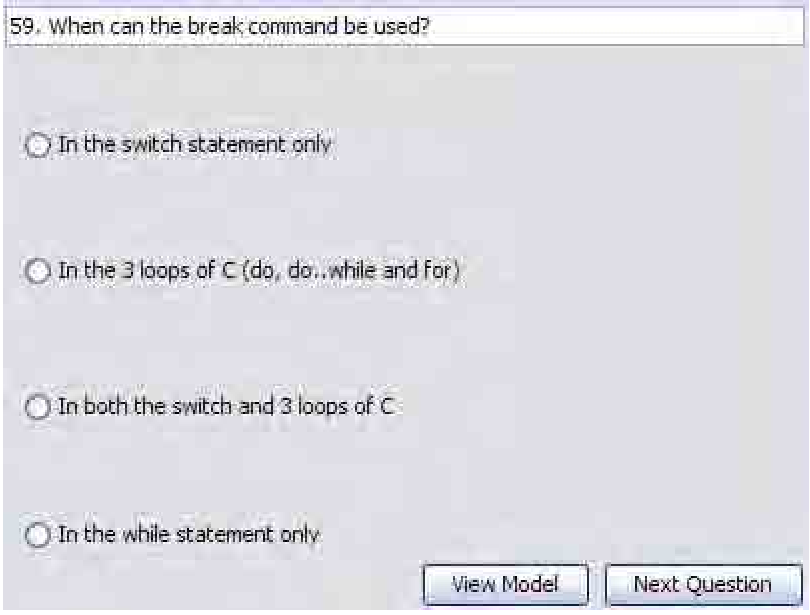
58. What is the purpose of the break command?

○ Exit from a loop

○ Exit from a switch

○ Exit from both a loop and switch

○ Exit from a while loop only



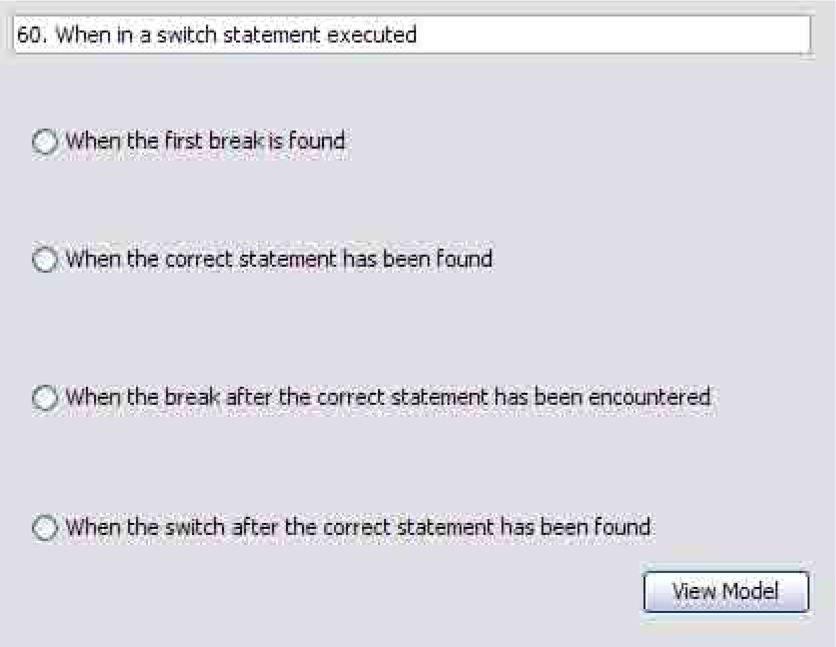
59. When can the break command be used?

○ In the switch statement only

○ In the 3 loops of C (do, do..while and for)

○ In both the switch and 3 loops of C

○ In the while statement only



60. When in a switch statement executed

○ When the first break is found

○ When the correct statement has been found

○ When the break after the correct statement has been encountered

○ When the switch after the correct statement has been found

Chi- Squared statistical analysis calculations – perceived ability and expected ability

Category	Actual Ability	Perceived Ability (Expected)
2 or less	15	8
3	4	7
4 or more	1	5

The category was determined based on the answer the participant gave to initial question in the system, how good they believed they were at C programming, this was a 5 point scale.

(H0) Null Hypothesis = There is no difference between each participants' actual ability and their perceived ability.

(H1) Alternative hypothesis: There is a difference

There are 2 degrees of freedom (3 groups – 1)

Test statistic = $\sum \frac{(Actual - Expected)^2}{Expected}$

Test statistic = ((15-8)^2 / 8) + ((4-7)^2 / 7) + ((1-5)^2 / 5) = 10.6107

(Chi-squared) $\chi^{2}_{2}, 0.05 = 5.991$

From this we note that less than 5% of the distribution lies beyond the value 5.991, as 10.6107 > 5.991, therefore we reject the null hypothesis in favour of the alternative hypothesis. We conclude that there is a significant difference between perceived ability and actual ability

Chi- Squared statistical analysis calculations – what the model does like and should like

Category	Model should look like (Actual)	Model did look like (Expected)
1 & 2	15	4
3 & 4	5	8
5 & 6	0	8

These categories were based on what the learner model of each of the participants looked like (after editing) and should have looked like (edited appropriately). The categories go up to six, as there were six concepts the learners could have displayed knowledge in.

(H_0) Null hypothesis: there is no difference between what the model does look like and what it should look like

(H_1) Alternative hypothesis: there is a difference

There are 2 degrees of freedom (3 groups – 1)

Test statistic = $\sum \frac{(Actual - Expected)^2}{Expected}$

Test statistic = $((15-4)^2/4) + ((5-8)^2/8) + ((0-8)^2/8) = 39.375$

(Chi-squared) $\chi^2_{2,0.05} =$ 5.991

From this we note that less than 5% of the distribution lies beyond the value 5.991, as 39.375 > 5.991, therefore we reject the null hypothesis in favour of the alternative hypothesis. We conclude that there is a significant difference between, what people believe their model should look like and what it actually does look like.

A position paper, where work from the thesis is included

EC-TEL 2010: Workshop on Technology-Enhanced Formative Assessment (TEFA)

A Role for Open Learner Models in Formative Assessment: Support from Studies with Editable Learner Models

Norasnita Ahmad, Mark Britland, Susan Bull and Andrew Mabbott

Electronic, Electrical and Computer Engineering

University of Birmingham, UK

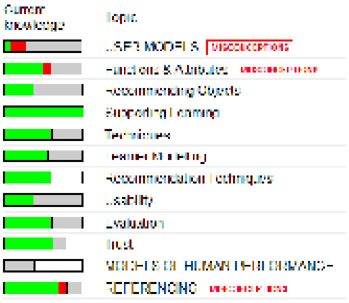
Position Statement: Summary

This position statement builds on arguments for developing open learner models to provide formative assessment opportunities and promote learner reflection. We suggest that students do use open learner models for such metacognitive activities, illustrated by their behaviour when using learner models that they can directly edit. We conclude that, in addition to theoretical justifications for open learner models to support formative assessment, there is initial evidence showing students will not simply change their model to reveal what they want it to show – even if this could gain them additional or ‘easy’ marks. Instead, students will use an open learner model to help them plan and monitor their learning.

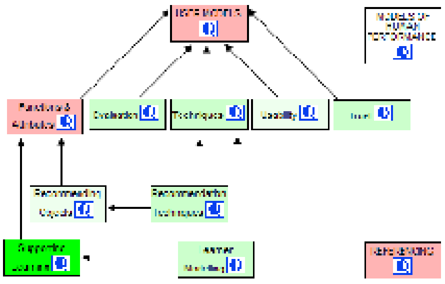
Open Learner Models

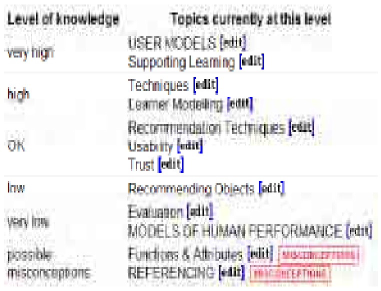
Adaptive learning environments individualise the interaction to suit the learner’s educational needs. This personalisation is achieved by dynamically modelling the learner’s understanding as revealed (or inferred) from their actions in the environment. In most adaptive systems, the learner model is used only or primarily for this purpose, and so is not available for user viewing. ‘Open learner models’ are learner models that are accessible for user inspection in an understandable form, to help the learner identify their current knowledge or understanding of a subject, sub-topic or specific concept. They may then use this information to help them reflect on their knowledge, identify gaps in their understanding and plan their learning (Bull & Kay, 2007). The open learner model thus provides the user with additional information about their learning progress that is not usually available to them.

Learner models can be displayed to users in a variety of ways. We here illustrate open learner model interfaces using two ‘learner model views’: skill meters (left) and a structured learner model presentation which shows relationships between topics (centre). Colour is used to indicate strength of understanding (green), misconceptions (red), general difficulties that are not identified as specific misconceptions (grey), and areas that have yet to be attempted (white). Clicking on ‘misconceptions’ links in these (and the other) learner model views leads to simple statements of the learner’s misconception, as a starting point for them to work out their difficulties. (For example, in a course about user modelling: “You may believe that adaptive systems do not have any drawbacks”.)



Current knowledge	Topic
	USER MODELS MISCONCEPTIONS
	Functions & Attributes MISCONCEPTIONS
	Recommending Objects
	Supporting Learning
	Techniques
	Learner Modelling
	Recommendation Techniques
	Usability
	Evaluation
	Trust
	MODELS OF HUMAN PERFORMANCE
	REFERENCING MISCONCEPTIONS





Level of knowledge	Topics currently at this level
very high	USER MODELS [edit] Supporting Learning [edit]
high	Techniques [edit] Learner Modelling [edit]
OK	Recommendation Techniques [edit] Usability [edit] Trust [edit]
low	Recommending Objects [edit]
very low	Evaluation [edit] MODELS OF HUMAN PERFORMANCE [edit]
possible misconceptions	Functions & Attributes [edit] MISCONCEPTIONS REFERENCING [edit] MISCONCEPTIONS

Editable Learner Models

Editing the learner model (e.g. above right) may be helpful in situations where a learner has additional information about their knowledge, for example: in cases where understanding has not yet been demonstrated during an interaction (such as existing background knowledge, understanding gained as a result of attending a lecture or reading course notes/other information); or where the student is aware of having forgotten information that they had previously known. This accords with the notion that learners should have control over, and responsibility for their learning, and hence their learner model (Bull & Kay, 2007). We here briefly present the results of three studies undertaken with university students:

(i) A lab-based study with 20 participants, where some of the contents of the learner model were automatically changed before the learner model was presented to the learner. The purpose was to identify whether students would edit the inaccuracies introduced into the representations in their model, while not editing the model data that accurately reflected their understanding.

(ii) A study of a deployed open learner model with 135 students over two terms, where the model inferences were not altered by the system in this way (i.e. the representations were assumed to accurately reflect the user's knowledge), but where the user could edit their model if they wished.

(iii) A study of a deployed open learner model with 18 students during one term, again where the model was not changed by the system before presentation, but where the user could edit their model up until the point that it was summatively assessed (the model contributed 5% to the final course mark).

In study 1 students generally edited their learner model when a misconception they did not actually hold, was inserted before the model was displayed. However, they did not do so when additional correct beliefs were inserted into their model. They tended not to edit the accurate (i.e. unchanged) representations.

Study 2 showed very little model editing activity at any point during the deployment. Study 3 showed that, while some students edited their learner model at some points during the term, only one edited their model to show knowledge that was not verified, before the summative assessment deadline.

Discussion

Given the potential for open learner models to prompt reflection by showing users inferences about their knowledge, we believe this to be a useful approach to encourage metacognitive activities related to self regulation and planning, and to support formative assessment (see Bull & Kay, 2007). We have provided examples of cases in which open learner models are *editable* by the user. Although the lab-based study showed a tendency for students to remove erroneously represented misconceptions from their learner model, they did not notice, or chose not to remove knowledge that was automatically added – i.e. knowledge they did not actually have. Nevertheless, the accuracy of the edits that they did make, seems confirmed. In situations of deployed open learner models, students generally made few edits. This was found both in a study with a large number of students who were optionally using the system for formative assessment only; and in a smaller scale study where the learner model was also summatively assessed. It is this latter case that is particularly interesting: it suggests that the perceived benefits of formative assessment supported by an open learner model outweighed the temptation to simply edit the learner model to gain course credit. Therefore, in addition to theoretical reasons for providing open learner models to support formative assessment, we believe that students themselves see them as beneficial, and will use an open learner model appropriately to support their independent learning.

Reference

Bull, S. & Kay, J. (2007). Student Models that Invite the Learner In: The SMILI Open Learner Modelling Framework, International Journal of Artificial Intelligence in Education 17(2), 89-120.