Unit 205 Create Software Components using Java Level 2 (Core)

Rationale

The aim of this unit is to enable candidates to understand the principles required to create applets using the Java programming language. Candidates will develop the skills required to create and test software components or small software systems to solve a given problem.

This unit has been written to comply with Java JDK1.2 (Java 2). However, this does not preclude centres from using Swing components.

There are 4 outcomes to this unit. The candidate will be able to:

1. manage the development environment
2. use components to create a Graphical User Interface (GUI)
3. create code for a specified software component
4. test a software component and produce printed output.

Guided learning hours

The recommended guided learning hours for this unit are 90 hours

Connections with other awards
NVQ links

Developing IT Programs Level 2
216.1 - Assist the creation of software
216.2 - Assemble and test software components

Key Skills links

Communication	C3.2
Application of Number	N1.1
Information technology	None
Working with others	None
Improving own learning	LP3.1, LP3.2, LP3.3
Problem solving	PS3.1, PS3.2, PS3.3

Assessment
Assessment will be by means of a **set assignment** covering practical activities, and a **multiple choice test** covering underpinning knowledge.

Outcome 1: Manage the development environment

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. access the development environment 2. create a source code file (**java**) for a java applet using a text editor 3. compile a source code file into a bytecode file (**class**) 4. resolve syntax errors flagged by the compiler 5. create an **HTML** file which contains a reference to a Java applet 6. run an applet using the appletviewer or a browser 7. exit the development environment.		
Underpinning knowledge **The candidate will be able to:** 1. describe the purpose and function of the following file types: • **java** • **class** • **html** 2. state the difference between a Java application and a Java applet 3. state the relationship between an applet, an HTML document and a browser.		

Outcome 2: Use components to create a Graphical User Interface (GUI)

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. use a layout manager 2. create and use methods to manipulate components 3. create and use methods to manipulate a dialog 4. set the attribute values of component objects 5. load, display and scale images 6. draw lines and shapes and use different text fonts and font styles 7. implement a listener to handle events generated by components.		

Underpinning knowledge
The candidate will be able to :

1. describe the different layout managers and how they organise components in a container:
    - flow
    - border
    - grid

2. describe components with which the user can interact via a mouse or keyboard:
    - Text field
    - Label
    - Text area
    - Check box
    - Radio button
    - Choice
    - List
    - Button
    - Frame
    - Menu

3. describe containers, Frames and Menus

4. state that a **Panel** is a container used for organising components

5. describe modal and non-modal Dialog

6. state that a listener is implemented to enable an object to become an event listener and be notified of an event.

Outcome 3: Create code for a specified software component

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. use comments to document code 2. use consistent indentation and presentation of code to improve readability 3. use the **import** statement to access predefined classes from packages 4. declare and use the data types: • **int** • **float** • **boolean** 5. declare and use constants and built-in predefined constants as appropriate 6. create new objects 7. use and manipulate string, font and color objects 8. create user-defined classes 9. use extends to create a user defined class 10. use operators: assignment operator **=** relational operators: **==, <, >, !=, <=, >=** arithmetic operators: **+, -, *,** / logical operators: **&&** (AND), **||** (OR), **!** (NOT) 11. use program constructs for iteration: • **for** • **while** • **do…while** 12. use program constructs for selection: • **if** • **if…else** • **switch** 13. use the break statement 14. modify colours using the RGB model 15. use the **paint**() method to draw an applet 16. use the **drawString**() method of a **Graphic** object to output text to the screen.		

Underpinning knowledge
The candidate will be able to:

1. describe in simple terms the operation of software where discrete sections of code run in response to user-initiated events

2. describe the syntax for comments;

3. state that a class is a template for an object

4. identify the structure of a class, its attributes and methods

5. describe the syntax for a method declaration, how its type is determined and its arguments specified

6. describe the logical and relational operators, the precedence rules for arithmetic and the effects of parenthesis

7. state limitations on the use of Java reserved words

8. describe the operation of iteration program constructs:
   - **for**
   - **while**
   - **do…while**

9. describe the operation of selection program constructs:
   - **if**
   - **if…else**
   - **switch**

10. state that the Abstract Windowing Toolkit (AWT) is a package of classes that implements most common User Interface (UI) components and also generates and manages events

11. describe the use of packages and the purpose of the import statement

12. describe applet security (read, write, delete, rename)

13. state that the **init**() method is used to load and initialise an applet

14. describe how the **stop**() method is called when execution of an applet is suspended and how the **destroy**() method is called when an applet is being removed from memory.

Outcome 4: Test a software component and produce printed output

	Candidate's signature	**Date**
Practical activities **The candidate will be able to:** 1. use test data to determine the expected results from a software component 2. compare the expected to the actual results and correct any errors 3. use available tools to identify errors 4. resolve logical and run-time errors found during testing. 5. provide evidence that the program complies with the specification 6. print the Graphical User Interface (screen images) 7. print listing of code.		
Underpinning knowledge **The candidate will be able to:** 1. describe and distinguish between syntax errors and logical errors 2. identify the cause of a run-time error 3. state the reasons for testing a software component prior to implementation 4. identify that testing for expected output can assist in determining whether or not a program is working correctly and conforms to the specification.		