

Sample assessment task

Unit 2, Outcome 1

Programming

Outcome 1

On completion of this unit the student should be able to design working modules in response to solution requirements, and use a programming or scripting language to develop the modules.

Task

For this Outcome, students create a portfolio of modules using a programming language.

The language used must be able to support object-orientated programming.

Programs are often made up of a number of independent modules. A module is considered to be one part of a program. Each module in the portfolio should focus on completing one specific task.

The teacher is required to present students with a list of solution requirements for each module. Students are required to design the module using appropriate design tools for functionality and appearance. Each module is then developed using an appropriate programming language. Finally, each module needs to be debugged and tested to ensure that it is working as expected.

All modules for the portfolio can be completed at the end of the unit. Alternatively, individual modules may be completed progressively throughout the unit.

Teachers have a choice on how the assessment task is completed. Having students complete all modules of the portfolio under test conditions is one option; having the students complete each module as an ongoing project is another.

Regardless of the method used to conduct the assessment task, the teacher must be able to authenticate that the work completed by each student is their own.

Requirements

Task 1: Manipulating text

Create a module that allows the user to input their name and their year of birth. When the program is executed, the age the user will turn this year is calculated, and a welcome sentence will be displayed.

For example, if the name of the user was Mark and the year 2000 was inputted, the following sentence would appear:

Welcome Mark, you will turn 16 this year.

Solution requirements

- 1 User inputs name and year of birth
- 2 Age is calculated
- 3 Welcome sentence is constructed
- 4 Welcome sentence is outputted

Steps

| | | |
|---|--|---------|
| 1 | Using an appropriate design tool create an algorithm showing how the module will function. | 4 marks |
| 2 | Using an appropriate design tool show how the module will appear. Annotate the design tool to show all the objects used. | 4 marks |
| 3 | Using an appropriate design tool show the variables to be used in the module. | 2 marks |
| 4 | Using a programming language develop the module based on the design specifications. | 6 marks |
| 5 | Create a test table that will check to see if all the solution requirements are working correctly. | 4 marks |

Task 2: Manipulating numbers

Create a program that allows the user to input two numbers. The total and the average of the two numbers are calculated and then displayed.

Solution requirements

- 1 User inputs two numbers
- 2 The total and average are calculated
- 3 The total and average are outputted

Steps

| | | |
|---|--|---------|
| 1 | Using an appropriate design tool create an algorithm showing how the module will function. | 4 marks |
| 2 | Using an appropriate design tool show how the module will appear. Annotate the design tool to show all the objects used. | 4 marks |
| 3 | Using an appropriate design tool show the variables to be used in the module. | 2 marks |
| 4 | Using a programming language develop the module based on the design specifications. | 6 marks |
| 5 | Create a test table that will check to see if all the solution requirements are working correctly. | 4 marks |

Task 3: Decisions

Create a program that allows the user to enter a mark. If the mark is less than 50, the mark is allocated the award of 'unsatisfactory'. Otherwise, the mark is allocated the award of 'satisfactory'. Finally, the award is displayed.

Solution requirements

- 1 User inputs a mark
- 2 Award calculated
- 3 Award displayed

Steps

| | | |
|---|--|---------|
| 1 | Using an appropriate design tool create an algorithm showing how the module will function. | 4 marks |
| 2 | Using an appropriate design tool show how the module will appear. Annotate the design tool to show all the objects used. | 4 marks |
| 3 | Using an appropriate design tool show the variables to be used in the module. | 2 marks |
| 4 | Using a programming language develop the module based on the design specifications. | 6 marks |
| 5 | Create a test table that will check to see if all the solution requirements are working correctly. | 4 marks |

Task 4: Loops

Create a program that allows the user to enter 4 numbers. After each number is entered the current total of the numbers is displayed. As each new number is entered the total updates to display the current sum of the numbers entered.

Solution requirements

- 1 User input numbers
- 2 Total calculated
- 3 Total displayed

Steps

| | | |
|---|--|---------|
| 1 | Using an appropriate design tool create an algorithm showing how the module will function. | 4 marks |
| 2 | Using an appropriate design tool show how the module will appear. Annotate the design tool to show all the objects used. | 4 marks |
| 3 | Using an appropriate design tool show the variables to be used in the module. | 2 marks |
| 4 | Using a programming language develop the module based on the design specifications. | 6 marks |
| 5 | Create a test table that will check to see if all the solution requirements are working correctly. | 4 marks |

Task 5: Read from file

Create a program that will read data from a file. Once the data is read from the file, it is then displayed using an appropriate object.

Solution requirements

- 1 Read data from file
- 2 Display data

Steps

| | | |
|---|--|------------------|
| 1 | Using an appropriate design tool create an algorithm showing how the module will function. | 4 marks |
| 2 | Using an appropriate design tool show how the module will appear. Annotate the design tool to show all the objects used. | 4 marks |
| 3 | Using an appropriate design tool show the variables to be used in the module. | 2 marks |
| 4 | Using a programming language develop the module based on the design specifications. | 6 marks |
| 5 | Create a test table that will check to see if all the solution requirements are working correctly. | 4 marks |
| | Total (across all tasks) | 100 marks |

Rubric

The rubric below can be used to allocate marks for each step:

| Step | Marks | | | | | | |
|-------------------------------|--|--|---|--|--|--|--|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Design tool for functionality | No algorithm is shown | Is present but is only partially attempted | Shows most of the detail required | Complete but not fully correct | Fully complete and correct | | |
| Design tool for appearance | No design for appearance shown | Is present but is lacking most detail | Is present and contains most details | Is complete with most objects identified | Contains all details required including identifying all objects used | | |
| Design tool for variables | No design tool for variables used is shown | Some variables have been identified | All variables have been identified | | | | |
| Programming languages | No evidence of a programming language used | Some attempt at coding has occurred | A number of lines of code within the module are shown | Most of the code required is shown | Code is complete but contains two or more errors | Code is complete but contains an error | Code is fully complete and working as expected |
| Testing | No evidence of testing is provided | One or two tests or pieces of test data have been identified | Most tests and and/or test data have been identified | All relevant tests and test data have been identified but no evidence is provided that the tests have been carried out | All relevant tests and test data has been identified including evidence that the tests have been carried out | | |