COM00141M

Department of Computer Science

ALGORITHMS AND DATA STRUCTURES

ASSESSMENT BRIEF

Assessment Author	Tommy Yuan
Assessment	Summative - Open
Release	Week 1
Submission	Tuesday Week 7 at 13:00 (UK time)
Feedback	Within 20 working days of submission
Weighting	20%

I. Module Learning Outcomes

At the end of this module, you will be able to:

1. Express a problem solution algorithmically using pseudocode
2. Analyse the time complexity of an algorithm
3. Construct computer programs to implement algorithms
4. Test a computer program against the specification.

This assessment will contribute to learning outcomes 3 and 4 for this module.

II. Practical Programming Assignment 2

Introduction

- You are to develop objects for a computer system.
- This assignment is divided up into separate parts to help you with your development

- Please refer to the mark scheme in section 3 below for mark weightings

III. Assessment Task/s

Part A

- Using Eclipse you should create a new System project.
- Within the project, create a System class
- Below is the class diagram for the System class:

System
-make : String -model : String -speed : int -memorySize : int //in MB -hardDiskSize : double //in GB -purchaseCost : double
+System(String,String, int) +setMemory(int) +setHardDisk(double) +setPurchaseCost(double) +getMake():String +getModel():String +getProcessorSpeed():int +displayDetails(): String +checkHDStatus(): String +goodMemorySize() : boolean +diagnoseSystem(): String

System(String, String, int)

- The constructor should initialise the make, model and processor speed of the system.

getMake():String

getModel():String

getProcessorSpeed():int

- Methods to return attribute values.

setMemory(int)

setHardDisk(double)

setPurchaseCost(double)

- These methods should set the corresponding attributes.

displayDetails(): String

- This should display all the details of the system. These details should be output as a **String**.

checkHDStatus(): String

- This method should check if the hard disk size is below 2 (GB). If so, it should return “low”, otherwise “OK”

goodMemorySize(): boolean

- This method should check if the ram memory size is below 128 (MB). If so it should return false, otherwise true

diagnoseSystem(): **String**

- This method should use both the checkHDStatus() and goodMemorySize() methods to return a **String** displaying appropriate messages as below:

Hard disk size = low
Memory size OK = false

Within your project create another class called **SystemTest**

- Create a '**main'** method, which should create a System object and test all the methods.
- You should 'set' the details within the code.
- Compile and run the **SystemTest** class **from the command line**
- Evidence your source code and output in your assignment report.

Note – Your program should run from the command line i.e. independently of Eclipse.

Part B – System properties

Java provides a System class with many useful static methods (e.g. System.out.println()).

One of the methods provided in the System class is ***getProperty(argument)***. Details of this and how it can be used can be found at:
http://java.sun.com/docs/books/tutorial/essential/environment/sysprop.html
Add a '**displaySystemProperties(): String'** method to your System class. This method should display, in a **String**, the following properties:

- Operating System Architecture
- Operating System Name
- Operating System Version
- User Account Name
- Java Version.

The method should also output a positive message if the Operation System equals 'Windows 10', a negative message if the Operating System equals 'Linux' and a neutral message for the rest of the operating systems.

Test the **displaySystemProperties()** method by calling it from the test class you wrote in part A.

Part C - Test class with textual menu

Amend your test class so that, after it initialises a System object, it presents the user with the following textual menu in the console (use as many options as you have implemented).

- Choice 1 – Print System Details
- Choice 2 - Display System Properties
- Choice 3 – Diagnose System
- Choice 4 – Set Details
- Choice 5 – Quit the program

This menu should be repeated until the user quits.

When Choice 4 is chosen, the memory size and hard disk size of the computer should be interactively set. E.g. prompt the user to input the size of the memory and Hard disk.

Part D – Upgrade Part A – C to a GUI interface

Within your project create another class called **SystemTestGUI**

- Create a '**main'** method, which should create a System object and test all the methods.
- Create a **graphical menu** based on the structure of your textual menu in Part C, providing appropriate user interaction for input and output.

You should explore the range of GUI components available to you. This may be a simple series of screens using JOptionPane or JavaFX (*Figure 1*) or a hierarchical navigation structure with appropriate layouts (*Figure 2. Advanced GUI with multiple screens and elements.*). *Note: these figures are only provided as examples, your design does not need to match them.*

Make sure you carefully consider what type of data (information) you want the user to input and that your selected components support this (consider the data types in the class diagram).

Credit will be given for appropriateness, diversity (hierarchical navigation, choice of components where applicable) and clarity of arrangement (intuitive interface, clear indication of input required, concise display of output).

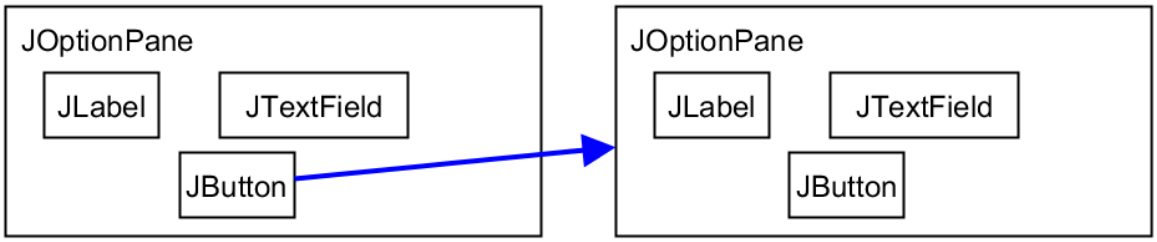

Figure 1. Simple GUI

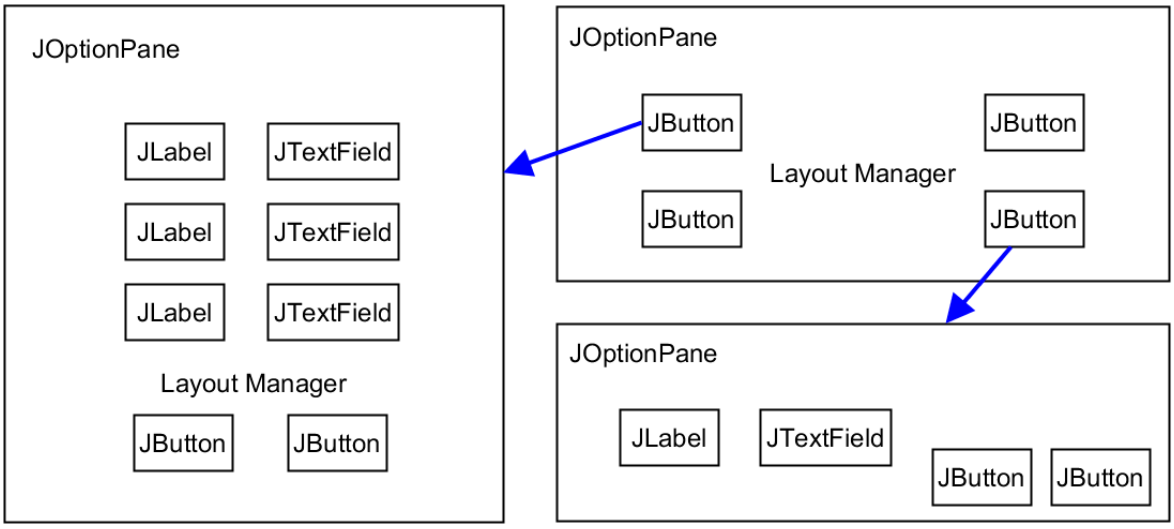

Figure 2. Advanced GUI with multiple screens and elements.

IV. Deliverables

Please submit the following:

1 A single report in **PDF (or word)** format, which contains evidence of **ALL testing** and outputs (screen shots and snippets of code).

2 A **single zip file** that contains all your java source code files – for Part A,B,C and D and the two test files **SystemTest** and **SystemTestGUI**

V. Marking Criteria

A marking scheme and assessment criteria can be found below:

The marks for each section are detailed below. The maximum mark is 100. This will be scaled to 20% for the assignment overall.

The following contributes to the learning outcomes 3 & 4 as outlined earlier in section I		
Section	**Criteria**	**Available marks**
A. **Implementation** Attributes and method declarations	Source code evidence for appropriate attributes, method signatures and data types.	**15**
Instantiation	Evidence of System and Test objects being instantiated and initialised correctly.	**5**
Test Class	Test class used to instantiate and test object.	**5**
Section	**Criteria**	**Available marks**
B. **System Properties**	Appropriate display of System Properties with messages.	**10**

C. **Textual Menu**	Functional textual menu.	**20**
D. **Graphical Menu**	Functional GUI components to prompt for and display details.	**10**
GUI Design	Appropriateness, diversity and clarity of arrangement of GUI components.	**5**
Functionality/Testing	Evidence that all methods from Part A to D have been tested and function correctly (marks scaled accordingly)**.**	**20**
General	Source code is nicely presented and self-documenting, with comments. Appropriate class, method and attribute naming. – follows coding convention with appropriate indentations.	**10**
	Total	**100**

NOTE: Non submission of your java source code files will result in 0 (zero) marks for the assignment **as all java files are tested to verify the content in your report.**

VI. Grading

The pass mark for postgraduate modules is 50%. For more information about grades and assessment criteria, please review the 'Assessment and award' section of the York Online Handbook which can be downloaded from the **Orientation Module**.

VII. Assessment submission

You will submit your assessments in the 'Assignments' area of the module in Canvas. Please check your canvas module for the specific submission date for this assignment.

For general assessment guidelines consult your Canvas Module, Orientation Module and for Academic Regulations the University of York's website.

Any queries regarding the details of your assessment should be directed to your module tutor.

Any queries regarding assessment procedures should be directed to studentsuccess@online.york.ac.uk.