Department of Computer Science

ALGORITHMS AND DATA STRUCTURES

I. Module Learning Outcomes

- **MO1**: express a problem solution algorithmically using pseudocode
- **MO2**: analyse the time complexity of an algorithm
- **MO3**: construct computer programs to implement algorithms
- **MO4**: test a computer program against the specification.

This assessment will contribute to learning outcomes **MO3** and **MO4** for this module.

II. Assessment Brief

You are required to develop a Java program with a **graphical user interface (GUI)** to render 2D objects based on user input, visualise some transitions between states and demonstrate how you've preserved states between transitions. You must use the same development environment as the one used in this module (**Java 11** and **JavaFX)** to implement the logic and GUI components.

The program should have the following functionality:

1. **[Component 1]** An interactive component which has the following actions associated to it; the component can accept user input, where assuming the input is valid, a shape will be displayed on the main screen. It is assumed that the program will eventually support a large variety of shapes. The program should currently accept the following shapes as input: **semi-circle**, **rectangle**, **pentagon**.
   1.1. You need to ensure that the user input is validated, and feedback is given when invalid input is entered.
   1.2. The program should only allow the user to enter one valid shape at a time.
   1.3. The main screen should only display one shape at a time.

2. **[Component 2]** An interactive component which has the following actions associated to it; the component must accept user input, where assuming the input is valid, the currently displayed shape will be filled using the colour of the valid input. Again, it is assumed that the program will eventually support a large variety of colours. The program should currently accept the following colours as input: **orange**, **blue**, **grey**.
    2.1. You need to ensure that the user input is validated, and feedback is given when invalid input is entered.
    2.2. The program should only allow the user to enter one valid colour at a time.
    2.3. The currently displayed shape should only be filled with one colour at a time.

3. Any potential error states should not crash the program or generate generic error messages. The program should clearly indicate to the user that an error has occurred, with specific information on why the error occurred and what valid action(s) are available.

III. Assessment Tasks

Given the brief stated above, your tasks are as follows:

- Create a Java class called ***Shapes.*** Implement the functionality from the brief stated above, ensuring that you have covered all points.

- Create a report with the following sections (a report template is provided at the submission point):

    1. **Implementation:** Outline of your approach taken to meet the functionality specified in the brief, with justifications and supporting evidence e.g. source code evidence for appropriate attributes, method signatures and data types. Design and justification of GUI components for prompting and displaying details.

    2. **Functionality and Testing:** Outline of your strategy with justifications and supporting evidence to demonstrate that **ALL** of the functionality specified in the brief has been tested. You also need to show that the *Shapes* class can be compiled and run **from the command line**.

    List of references: Provide a correctly structured list of references to all the resources used for this development and report. Your responses should be appropriately supported by references to the literature and relevant resources using the Computer Science

Report page limit: 6 pages total, all text and images inclusive but excluding list of references. Excess pages will not be marked.

IV. Deliverables

Please submit the following:

1. A single **report** in PDF (or word) format. **(Page limit: 6 pages, all text and images inclusive but excluding list of references. Excess pages will not be marked).**
2. A single **Java file** containing your program source code (.java extension) e.g. *Shapes.java*.

You should not be identified anywhere on your submission.

General submission guidelines:

- Present your answers on A4 pages, with a minimum 12 point font (14 point for headings), minimum 120% line spacing (what Word calls “Multiple 1.08”), and margins of at least 2 cm on either side.
- All images must be of a size/quality that the details are viewable within the document. Images will not be scaled to identify blurred or small text.

V. Marking Criteria

The marks for each section are detailed below. The maximum mark is 100. This will be scaled to 30% for the assignment overall.

Learning Outcome	Section/Task	Criteria	Available marks
MO3	1. Implementation	Outline of your approach taken to meet the functionality specified in the brief, with justifications and supporting evidence e.g. source code evidence for appropriate attributes, method signatures and data types. Design and justification	

		of GUI components for prompting and displaying details.	
		Functionality approach/justification: Component 1	10
		Functionality approach/justification: Component 2	10
		GUI design approach/justification: Component 1	10
		GUI design approach/justification: Component 2	10
MO4	2. Functionality and Testing	Outline of your approach taken, with justifications and evidence that all functionality in the *Shapes* class and the options in the GUI have been tested and that they fully meet the requirements of the brief.	
		Overview of strategy for testing/demonstration of functionality	10
		Testing/evidence of functionality: Component 1	10
		Testing/evidence of functionality: Component 2	10
		Testing/evidence of error handling: Component 1	10
		Testing/evidence of error handling: Component 2	10
MO3	General	Source code is well 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 file (.java extension) will result in 0 (zero) marks for the assignment **as all Java files are tested to verify the content in your report, and must compile and run from the command prompt using the version of Java specified in this module.**