SCHOOL OF ENGINEERING AND TECHNOLOGY

COURSEWORK FOR THE
BSC (HONS) INFORMATION TECHNOLOGY; YEAR 1
BSC (HONS) COMPUTER SCIENCE; YEAR 1
BSC (HONS) INFORMATION TECHNOLOGY (COMPUTER NETWORKING AND SECURITY); YEAR 1
BSC (HONS) SOFTWARE ENGINEERING; YEAR 1

ACADEMIC SESSION 2023; SEMESTER 2,3,4

PRG1203: OBJECT ORIENTED PROGRAMMING FUNDAMENTALS

DEADLINE: 31 JULY 2023 11:59PM (Monday)

INSTRUCTIONS TO CANDIDATES

- This assignment will contribute 20% to your final grade.
- This is a group (maximum 5 students) assignment

IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

Any work submitted after the deadline, or after any period of extension granted shall be marked as a Fail or awarded a zero.

Academic Honesty Acknowledgement

"I ....., ..., ..., ..., ... (student name). verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer student handbook undergraduate programme)* for any kind of copying or collaboration on any assignment."

....., ..., ..., ….., …... (Student's signature / Date)

Group Number: _____

Team Members:

No	Name	Student ID	Contribution %
1			
2			
3			
4			
5			

Marking Scheme

Criteria	Reference Marks		Marks	Remarks
Design (10%) Implement good object-oriented design in solving the problem, with high modularity, maintainability and reusability. Able to identify appropriate classes and their relationships, complete the classes with appropriate attributes and methods. The design is well presented in UML class and class relationship diagrams.	**10**	**Excellent**		
	7-9	**Good**		
	4-6	**Average**		
	1-3	**Poor**		
Coding (5%) Fulfil all the functionalities and align to the design you have presented in the UML diagrams. Follow the best programming practices, such as naming convention, indenting, code structure, optimisation, with appropriate exception handling. Good user-friendliness.	**5**	**Excellent**		
	4	**Good**		
	2	**Average**		
	1	**Poor**		
Additional Functionality (5%) Add at least one additional enhancement or functionality to your program. Explain the rationale and reasoning by providing justification that supports the decision.	**5**	**Excellent**		
	4	**Good**		
	2	**Average**		
	1	**Poor**		
TOTAL	**20**			

Tasks and Deliverables

You are required to upload the report and source code to the eLearn. Only the team leader needs to do the submission.

The submission should include:

- Cover page with the team group ID, team members name and student ID
- Class and class relationship diagram for all the classes in the system
- Screenshots with description to demonstrate the test results
- Explanation or justification of the additional functionality
- Source code (upload the zipped project folder)
  Note: Make sure all the java source files (*.java) are included. Submission with only the *.class file will be given zero mark.

Boat Racing Game

You are required to build a simple game 'Boat Race' in Java program that fulfil the below requirements. Analyze and develop the Java program as per described using the Object-Oriented design. You should design your program for optimum maintainability and reusability with the best practices of object-oriented techniques you have learnt. You also need to document your design using the UML class and class relationship diagrams.

The game rules:

- The game is a two players game. At the beginning of the game, each player will be allocated with a boat. During the game, the players take turn to throw the dice (you can use the random function to generate the random dice number) to decide how many steps should the boat move forward.
- The river can be visualised as 100-columns track as below, which is filled with random number of traps(#) and currents(C).

- Once the game started, all the traps and currents will be scattered randomly in the river. Some currents are stronger than the others, so as the traps. The stronger current or trap will make the boat moves more steps forward or backward. When boat hits the trap, the boat will need to move backward x number of steps, when the boat hits the current, it will move forward x number of steps. The boat should not be allowed to move beyond the river's boundary.

- Game will end when either player's boat reaches the end of the river. Display the location of the boats after every move.

When the game starts, display the Top 5 scores and ask the player for the name (short name with one word). You should count the total turns that each player takes in the games. When the game ended and the score of the player is within the top 5 scores, store the player's score and name in the 'TopScore.txt' text file. The list should be ordered by score in ascending order.

Tips: You can add any additional attributes to the objects in this game which you see fit

Additional Functionality:
Design and develop one additional function that may help to improve the game you have developed above.