STUDENT NAME:_____

# THE UNIVERSITY OF NEW SOUTH WALES, CANBERRA

# AUSTRALIAN DEFENCE FORCE ACADEMY

SEMESTER 2 – Examination

November 2020

| | |
|---|---|
| **Course Code:** | ZEIT1102 |
| **Course Name:** | Intro to Programming |
| **Paper No.:** | Final Exam |
| **Time Allowed:** | 2 Hours |
| **No of Questions on Paper:** | 9 Questions |
| **No of Questions to be Answered:** | 9 of 9 Questions |
| **Maximum total marks for exam:** | 100 |

**All questions are NOT of EQUAL value.**

**Answers must be submitted by uploading a pdf to the 2020 assignment box on moodle.**

**Examination Materials:**

**Candidate may bring into room**

- any non-electronic material

**Q1.** Consider the class represented by the following UML. For it, the getFastestTime() method should return the fastest of the two time attributes, while the toString() method should return the name, the fastest of the two times, and any associated formatting or text literals - it must not return the slowest time. Write those two methods.

| Racer |
| --- |
| private String name<br>private float time1<br>private float time2 |
| Racer(String, float, float)<br>setTime1(float)<br>setTime2(float)<br>String getName()<br>float getTime1()<br>float getTime2()<br>float getFastestTime()<br>String toString() |

[Q1 Total 10 marks]

**Q2.** Write a main method that creates two Racer objects, one for Bob, with times of 3.6 and 2.7. The other should be for John, but his times should be read from the user. The main method should then print all the details of the Racer object with the fastest fastest time. If both fastest times are the same, then a message indicating this should be printed, as well as the details of both Racer objects.

[Q2 Total 10 marks]

**Q3.** Write code to implement a class called RelayRacer. This class should be a subclass of Racer. However it should, in total, be capable of storing four names and four times. Write the methods for this class, such that

- it is possible to specify all its attributes as part of its constructor
- there are appropriate set and get methods for all times
- it has a getFastestTime method that returns the sum of all its `time' attributes
- it has a toString method that returns all the names, and the sum of all the 'time' attributes - it must not return the individual times.

[Q3 Total 18 marks]

**Q4.** Consider the following RacerStats interface

```
public interface RacerStats{

 float fastestTime();

 float[] allTimes();

}
```

*fastestTime()* is meant to return the fastest time, while *allTimes()* is meant to return an array that contains all the individual times.

Briefly explain, but don't write, what would have to be done to the Racer class, if it implemented this interface?

[Q4 Total 5 marks]

**Q5.** Write code for the static method fastestOfAllTimes(). This method should be passed an array of RacerStats objects. It should return the object, with the fastest individual time. You may assume that every location in the array holds either a RacerStats object or is null, and that the array contains at least one object. You may also assume that no two RacerStats objects have the same fastest time.

[Q5 total 13 marks]

**Q6.** Consider the following exception

```
public class IncompleteTimes extends Exception
{
  IncompleteTimes(String message)
  {
    super(message);
  }
}
```

Write a new averageTime() method for RelayRacer, such that if any of the times are not greater than 0, then the method throws an IncompleteTimes exception.

[Q6 Total 10 marks]

**Q7.** Consider the array, {3,6,2,5,1}. Assuming that it will be sorted in ascending order, using an insertion sort. For that sort, show the state of the array after each time it changes.

[Q7 Total 8 marks]

**Q8.** What would it mean if the class Racer was made abstract. Why, or why not, would this be done?

[Q8 Total 6 marks]

**Q9.** Write a method that can be passed any number of floats as multiple actual parameters. If less than 2 floats are passed, it should raise an IncompleteTimes exception. If 2 or more floats are passed, it should return the 2nd largest of all of them, i.e. if 3, 5, 9, 0, 2 and 11 are passed, then 9 should be returned. You may assume all passed floats are unique (i.e. there are no duplicate values).

[Q9 Total 20 marks]