BINDURA UNIVERSITY OF SCIENCE EDUCATION
FACULTY OF SCIENCE AND ENGINEERING
DEPARTMENT COMPUTER SCIENCE
BSc HONS DEGREE IN COMPUTER SCIENCE/INFORMATION TECHNOLOGY

CS203/_SWE211/_EEE2203: OBJECT ORIENTED PROGRAMMING II

DURATION: 3 HOURS TOTAL MARKS: 100

INSTRUCTIONS TO CANDIDATES

Answer all questions.
The paper consists of Section A (Theory) and Section B (Practical)
Section A carries 40 marks
Section B carries 60 marks
All programs to be written in Java

Section A: Theory

Question 1

a) A complete Java program may use the same name for several different methods or variables. Java has a number of features that allow the user to prevent such re-use of names from causing chaos. Describe these under the headings:

i. scope rules within individual functions; [4]
ii. visibility of method names within classes, and the effects of inheritance; [4]
iii. avoiding ambiguity when referring to the names of classes. [4]

b) Explain how the word "protected" is used in Java, commenting about when and why a programmer might use it rather than the alternative legal words that can appear in similar places. [6]

c) Why will you hardly ever use "protected" in small programs that you write and run for yourself? Explain why. **[2]**

Question 2

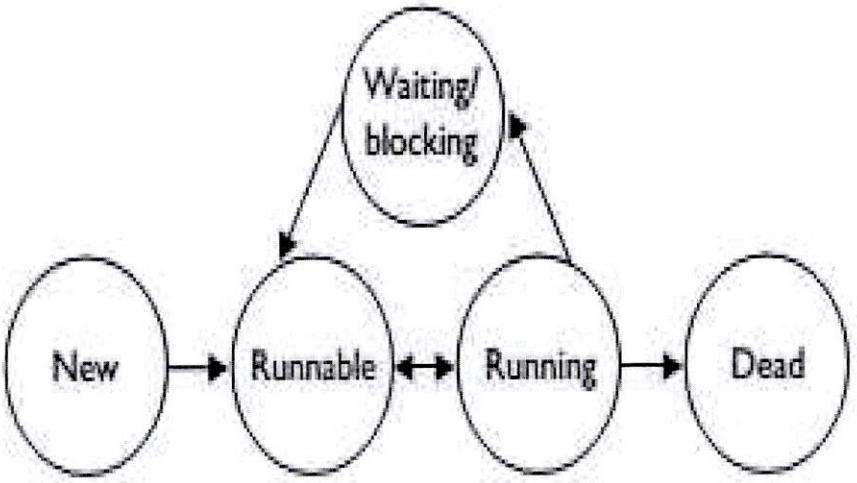

Figure 1: thread states

a) Describe the thread states in figure 1 and explain how they change from one state to another? **[12]**

b) Illustrate a real life scenario which can be modelled using threads. **[6]**

c) Explain any one disadvantage of using threads in java programs. **[2]**

Section B: Practical

Create a folder on the desktop and name it using your registration number and course code. Save all you work in this folder.

Question 3

The following class diagram in figure 2 represents a partial design for a major sporting event, such as the Olympic Games:

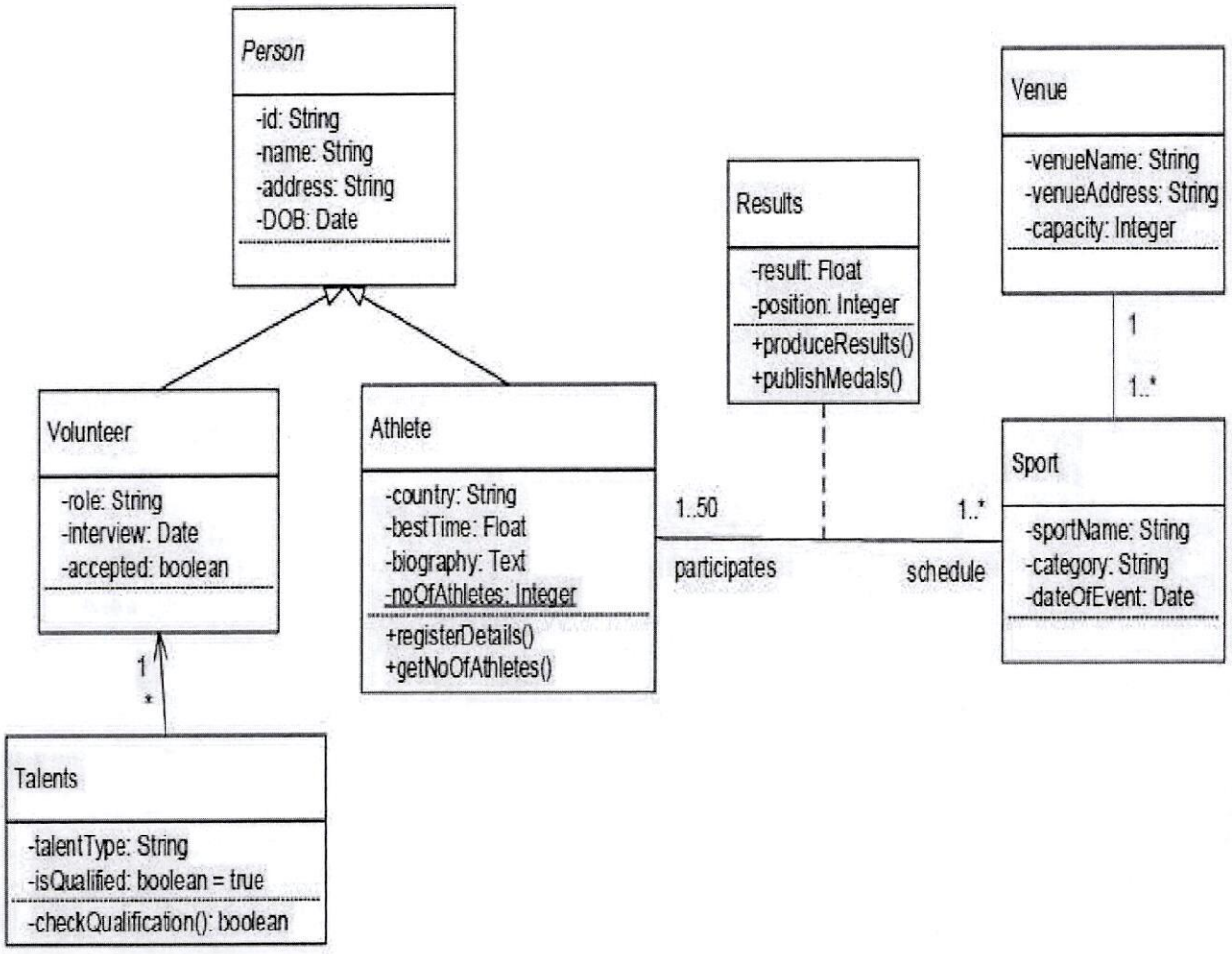

Figure 2: UML class diagram

Write Java versions of the all the classes in the diagram, implementing the variables, methods and relationships shown in figure 2. Include a suitable constructor in each class. **[20]**

Question 4

Write a complete java program which contains a static method named computeGrade that accepts as its parameter a Scanner for an input file whose data represents a student's grades for tests and assignments. Your method should compute the student's overall grade percentage from the total points earned for all tests and assignments versus the total points possible for all tests and assignments. The input consists of a series of one or more score records. Each score record consists of three tokens, where the first is the name of the assignment or test, the second is the number of points the student earned and the third is the number of points possible.

For example, if the input file contains the following text:

```
homework1 40 57 test1 78 100 test2 67 80
```

Your method would produce the following output. Notice that the grade percentage is either truncated or rounded to an integer.

```
homework1: 40/57
test1: 78/100
test2: 67/80

grade: 78%
```

The 78% comes from adding (40 + 78 + 67) and dividing by (57 + 100 + 80). **[20]**

Question 5

a) Write a compete program which contains a method that takes a LinkedList of Integers as its only parameter and returns another LinkedList which is the "partial sums" of the input LinkedList. Specifically, the first entry of the output list is the same as the first entry of the input list. The second entry of the output list is the sum of the first two entries of the input list. The third entry of the output list is the sum of the first three entries of the input list, and so on. For example, if the input LinkedList looks like,

```
[10, 5, 2, 7, 12, 23, 45, 17]
```

then the output LinkedList would be,

```
[10, 15, 17, 24, 36, 59, 104, 121]
```

[15]

b) In your class above, add another method that takes a LinkedList of Integers as its only parameter and returns another LinkedList which is the "partial sums" of the input vector, this time in reverse, in other words starting from the last element. **[5]**

END OF PAPER