University of Glasgow

Wednesday 27 April 2016
2:00 pm – 4:00 pm
(Duration: 2 hours)

DEGREES OF MSc in Information Technology, MSc in Software Development

PROGRAMMING

(Answer all 5 questions.)

This examination paper is worth a total of 75 marks

The use of a calculator is not permitted in this examination

INSTRUCTIONS TO INVIGILATORS
Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.

This page is intentionally left blank

1. Consider the following boolean variables:

```
boolean test;
boolean a = true
boolean b = false;
```

(a) Write the value of `test` after the following statement: `test = a && b`.

[2]

(b) Write the value of `test` after the following statement: `test = a || b`.

[2]

(c) Write the value of `test` after the following statement: `test = a && !b`.

[2]

(d) Write the value of `test` after the following statement: `test = !a || b`.

[2]

(e) Write the value of `test` after the following statement: `test = !(a && b).`

[2]

2. Write a helper method `factorial` that takes as input an integer number `value` and returns its factorial. The factorial of an integer `n` is the product `1*2*3*...*n` (for example, the factorial of `5` is `1*2*3*4*5=120`).

[10]

3. Consider two integer variables initialized as follows:

```
int a=9;
int b=6;
```

(a) Write what the program prints on the screen when the following statement is executed:

```
String output=String.format("%d/%d=%d",a,b,a/b);
System.out.println(output);
```

[2]

(b) Write what the program prints on the screen when the following statement is executed:

```
String output=String.format("6/9=%1.1f",a/(float)b);
System.out.println(output);
```

[2]

(c) Write what the program prints on the screen when the following statement is executed:

```
String output=String.format("%02d/%02d=%02d",a,b,a/b);
System.out.println(output);
```

[2]

(d) Write what the program prints on the screen when the following statement is executed:

```
String output=String.format("%d/%d=%2.2f",a,b,a/(float)b);
System.out.println(output);
```

[2]

(e) Write what the program prints on the screen when the following statement is executed:

```
String output =
String.format("%1.1f/%1.1f=%d",(float)a,(float)b,a/b);
System.out.println(output);
```

[2]

4. Write a complete program (`main` method) to read in a series of real numbers from a file called `in.txt`, calculate their maximum and minimum values (`max` and `min`, respectively), and then print out the numbers, together with their weighting on `System.out`. Each number and weighting should be on one line. The `weighting` of a number `num`, with maximum and minimum values `max`, `min` is calculated by the following formula.

```
weighting = (num - min) / (max - min)
```

The numbers should be printed with a field width of `8` and a precision of `4`, while the weighting has a field width of `4` and precision of `2`. They should be separated by a single space.

The input will consist of a series of positive real numbers terminated by `-1.0`. There will be no more than `100` numbers in the input. You will need to store the numbers in an array so that you can loop through them again after you have calculated the maximum and minimum values. Your program should cope with special cases, using comments to describe each special case and how it is dealt with.

[20]

5. Define a new class called `Team` that has a name and three integer data members: `won`, `drawn` and `lost`. It should have accessor methods for these data members.

(a) Write a code fragment (not a complete program) that creates a `Team` object called `rons_team` and assign the values `"LaVitesse"` to the `name`, and `3`, `1` and `4` to the `won`, `drawn` and `lost` members.

[6]

(b) Now define a class called `League` that just contains an array of 20 `Teams`. It should have an `add` method that takes a `Team` as a parameter and adds it to the next available position in the array. It should have a `find` method that takes a `String`, the team name, and returns the team with that name or `null` if the team is not found.

[9]

(c) Write a method for class `Team` called `points` that returns the points scored by that team, calculated as 3 points for a win, 1 for a draw and none for a loss. `LaVitesse`, as defined above, would have a points total of 10.

[2]

(d) Finally write a method of `League` called `sort` that sorts the league into descending order of points. Do not use the Java library `sort`.

[8]