University of Glasgow

Wednesday 1 May 2019
9.30 am – 11.30 am
(Duration: 2 hours)

DEGREES OF MSc Information Technology, Software Development and IT Cyber Security

PROGRAMMING

(Answer all 5 questions.)

This examination paper is worth a total of 60 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 program:

```
public class ExamClass {
    public static String makeString(int a, double b) {
      String s = "";
      s = String.format("a is %d, and b is %5.2f", ++a, b);
      return s;
    }
    public static void main(String[] args) {
      int a = 4;
      double aDouble = 3.141529;
      String formatted = makeString(a,aDouble); // Line B
      System.out.println(formatted); // Line A
      System.out.println(a); // Line C
      System.out.println(b); // Line D
    }
}
```

(a) This code will not compile due to an error on one of the lines in main. Which line (A, B, C, or D), and why?

[2 marks]

(b) What do each of the `public`, `static`, and `String` mean in the declaration of `makeString`?

[3 marks]

(c) Write down precisely what is outputted to the console from the line marked A?

[2 marks]

(d) How would this change if `++a` was changed to `a++`?

[1 mark]

(e) How would you need to change the line marked B if `makeString` was not `static`?

[1 mark]

(f) What is the output of the line marked C?

[1 mark]

2. Consider the following program:

```
public class Student {
    private String name;
    private String matricNumber;
    private double gpa;
    public Student(String name, String matricNumber) {
      this.name = name;
      this.matricNumber = matricNumber;
    }
    public void setGpa(double gpa) {
      this.gpa = gpa;
    }
    public String toString() {
      return name + "(" + matricNumber + "): " + gpa;
    }

    Public static void main(String[] args) {
```

```
        Student s,t,u;
        s = new Student("Simon","0123456");
        Student t = new Student("Bill","9876543");
        System.out.println(s); // Line A
        t = s; // Line C
        t.setGpa(3.5);
        System.out.println(s); // Line B
    }
}
```

(a) How many Student references and objects are created?

[2 marks]

(b) Write the outputs at lines A and B. Explain your reasoning.

[4 marks]

(c) Is it possible to access the "Bill" object after line C? Explain your reasoning.

[1 marks]

(d) You would like to create a subclass of `Student` called `PGStudent`. This should have an additional `double` attribute (`projectGrade`) which should be set via the constructor. The `PGStudent` should display the `projectGrade` rather than GPA in its `toString` method. Write the `PGStudent` class noting down any changes you might need to make to the `Student` class.

[6 marks]

(e) Consider the following code. What is happening at line D? Would the two print lines produce the same output or not? Why?

```
PGStudent a = new PGStudent("Mary","1234",56.7);
Student b = a; // Line D
System.out.println(a);
System.out.println(b);
```

[3 marks]

(f) The teaching office would like a simple application (non-GUI) that can load, from a file, student details in the format:

```
name matric projectgrade
```

(i.e. each student is one row, and the values are separated by spaces). Sketch a method that can load this file, and populate an **array** of `PGStudent` objects. State any assumptions you make. Don't worry about handling exceptions. This is testing your progamming logic and not memory of particular Java classes – you will not be penalised for minor syntactical errors, or small errors in Java class names etc.

[4 marks]

3. Explain what makes GUI programs different to programs in which interaction is driven by a command line interface.

[10 marks]

4. What is the MVC design pattern, what problem does it address, and how does it address those problems? As part of your answer, sketch a diagram showing the main elements of a program using the MVC pattern.

[10 marks]

5. What is an *interface* in Java, and what is the main advantage interfaces gives to the programmer? In your answer, outline a brief programming example of the use and benefit of an interface. Also, in your answer, explain how *interface* is different to *class*.

[10 marks]