University of Glasgow

Monday 10 December 2018
1:00pm – 2:00pm
(Duration: 1 hour)

DEGREES OF MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

COMPUTER SCIENCE 2P: JAVA PROGRAMMING 2

Answer all 3 questions

This examination paper is worth a total of 50 marks.

The use of calculators 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.

1. Answer the following questions about Java language concepts: (20 marks total)

(a) State four differences between an **abstract class** and an **interface**. [4]

(b) If you call a method that throws a a **checked exception**, what are the two approaches for handling that exception? [2]

(c) What is the difference between the **static** keyword and the **final** keyword? [2]

(d) What is the role of the **generic parameter** in classes such as `ArrayList`? [2]

(e) Name three properties of an enumerated type (i.e., one declared with the `enum` keyword) that are not true of a normal, non-enumerated type. [3]

(f) Name and briefly describe the two basic classes involved in **file I/O** when using the `java.nio.file` package. [4]

(g) What is the difference between the types **`int`** and `Integer`? How can you convert between the two types? What is one reason to choose `Integer` over **`int`**? [3]

2. This question concerns the Java programming language. (15 marks total)

(a) The following Java method has no programming errors and will run correctly, but is badly written. Identify **five stylistic problems** with the method. [5]

```
public boolean My_Method(String[] input) {
  boolean result = false;

  for (int zzz = 0; zzz < input.length; zzz++) {
    String s = input[i];
    if (s.equals("")) {
      result = true;
    }
  }

  return result;
}
```

(b) For each of the following Java code fragments, indicate **exactly** what will happen when it is compiled and executed. If it produces output, show the exact output; if it runs but produces an error, specify the error precisely; if it will not compile, describe what the problem is.

(i)
```
import java.util.Objects;
public class B {
   public String field;

   public B (String s) {
      this.field = s;
   }

   @Override
   public boolean equals (B b) {
      return Objects.equals(this.field, b.field);
   }

   public static void main (String[] args) {
      B b1 = new B("one");
      B b2 = new B("two");
      System.out.println(b1.equals(b2));
   }
}
```

[2]

(ii)
```
public class A {
   public int i = 3;

   public A(int val) {
      int i = val;
   }
}

A a = new A(5);
System.out.println(a.i);
```

[2]

(iii)
```
int i = 5.0;
int j = 10.0;
System.out.println(i / j);
```

[2]

(iv)
```
try {
   throw new Exception("Hello!");
   System.out.println("World!");
} catch (Exception ex) {
   System.out.println(ex.getMessage());
}
```

[2]

(v)
```
// File A.java
public class A {
   public A(int value) {
      System.out.println("Value is " + value);
   }
}

// File B.java
public class B extends A {
   public static void main (String[] args) {
      A a = new A(10);
   }
}
```

[2]

3. This question concerns Java class design. (15 marks total)

First, read the following description of an employee record system.

> You are to design a class to model the employees of a company. Every employee has an employee number, a family name, and a given name.
>
> In addition, each employee has a basic hourly rate, a number of contracted hours per month, and a number of hours actually worked (in the current month). Hourly paid workers are paid 1.5 times their normal rate for any hours above their contracted number.

(a) Write a class definition for `Employee`, incorporating all of the attributes mentioned above. Be sure to use appropriate data types and access modifiers. You should also define a public constructor for `Employee`, which should initialise all fields – the initial value for "hours actually worked" should be zero. The constructor should check that the parameters are sensible, and should throw an `IllegalArgumentException` if it is given bad input. You do not need to define any instance methods. [7]

(b) Write a method `computePay()` that computes and returns the employee's pay for the current month based on the current values of the appropriate fields. [4]

(c) The company has decided to start dividing employees into two categories: **salaried** employees, who receive the same pay every month regardless of the number of hours worked, and **hourly** employees, whose pay is computed as above. Describe how you would modify your class design above to deal with this situation, including any possible new classes that might be introduced: you may illustrate your answer with fragments of Java code but this is not required. [4]

END OF QUESTION PAPER