# CS_M01 Distributed Programming in Java

***(Attempt 2 questions out of 3)***

## Question 1.

(a) Explain how to create a new thread of execution in a Java program, using the `Runnable` interface. **[3 marks]**

(b) The following code is a partially completed class for a 'single item' buffer, which might be used for passing data between two threads. The given code is not thread safe, however.

```
public class BufferOne {

   private boolean available = false;
   private int contents;

   public int get() {
      available = false;
      return contents;
   }

   public void put(int value) {
      contents = value;
      available = true;
   }

}
```

- (i) Give an example sequence of events to illustrate that the code is not thread safe. **[2 marks]**

- (ii) Alter the code to make it thread safe. **[3 marks]**

(c) Explain the Object–Orientated concept of *encapsulation.* Your answer should include references to data hiding and privacy modifiers. **[8 marks]**

(d) What is meant by a **static** field. Give an example of why you would declare a field to be static. **[3 marks]**

(e) Show the format of a **switch** statement. When would you use a switch statement and what advantages does it confer? **[6 marks]**

**Question 2.**

(a) What are the main differences between the protocols TCP and UDP? **[2 marks]**

(b)
- (i) Outline the steps involved in creating a TCP server using Java and sockets, clearly identifying the key classes and methods involved. You are not required to provide complete executable code, though you may if you wish. **[4 marks]**
- (ii) How would your answer to the previous question be different in the case of UDP? **[2 marks]**

(c) Describe, using a suitable example, what is meant by the Object–Oriented concept of **inheritance**. Explain the advantages of inheritance with regard to code re–use. **[8 marks]**

(d) What does it mean when a field is declared as **final**. Why would you want to make a field final? **[4 marks]**

(e) Explain the difference between method *overloading* and method *overriding* and give an example of when each might be used. **[5 marks]**

**Question 3.**

(a) Consider the following classes :

```
public class MyClass{
   private String id;
   private double cost;

   public MyClass(String id, double cost){
      this.id = id;
      this.cost = cost;
   }
}

public class AnotherClass extends MyClass{
   private String name;

   public AnotherClass(String name, String id, double cost){
      this.name = name;
      this.id = id;
      this.cost = cost;
   }
}
```

Explain why this code will not compile. Identify two ways in which the code should be altered to allow compilation. **[8 marks]**

(b) Explain the Object–Oriented concept of **polymorphism** using an ArrayList as an example. Your answer should make reference to subjects such as inheritance, casting, and static and runtime binding. **[8 marks]**

(c) Explain what **abstract** classes and **interfaces** are. What is the purpose of an interface and what advantages do we get from creating interfaces? **[4 marks]**

(d) Explain, with suitable examples, the use of the keyword **super**, with special reference to constructors in an inheritance hierarchy. **[5 marks]**