

## CS\_214 OBJECT TECHNOLOGY

*(Attempt 2 questions out of 3)*

### Question 1.

- (a) Explain what is meant by the O-O concept of **polymorphism** using an *ArrayList* as an example. Your answer should include discussion of topics such as *static* and *dynamic* typing, *casting* etc.

[10 marks]

- (b) Consider the following class :

```
public class MyClass{
    private String name;
    private int age;
}
```

Write two constructors for this class, one of which takes no parameters and sets the fields to sensible values, and one which takes values as parameters.

[5 marks]

- (c) Explain what is meant by the Java keyword **abstract**. Why can constructors not be declared abstract?

[7 marks]

- (d) Explain, using examples, what is meant by method *overloading*.

[3 marks]

## Question 2.

- (a) Discuss *encapsulation* of data in O-O languages. In particular you should contrast the O-O view of data with the procedural programming view. You should also explain the importance of encapsulation in software engineering.

[10 marks]

- (b) 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]

- (c) Explain, using examples, what is meant by method *overwriting*.

[3 marks]

- (d) Where is a field or method visible if it is declared as:

- protected
- public

[4 marks]

### Question 3.

- (a) Explain, using sample code and diagrams, what is meant by the O-O concept of **inheritance**. Explain how this concept helps with the software engineering goals of code re-usability and correctness.

[10 marks]

- (b) Assume that a class **Book** exists and that a collection of books are stored in another class called **Library**. The **Book** class has fields of *author*, which is a string and *cost*, which is a double. Look at the following code fragment from the class **Library** :

```
.....  
System.out.println(collection.get(i));  
.....
```

...where **collection.get(i)** returns (as an object) the  $i_{th}$  item in the collection. What sort of output would you expect from this code? Explain two methods of altering the code (in either class) so that the book's author and cost were printed.

[6 marks]

- (c) The **Integer** class has a **static** method *parseInt(String)*. What is a static method and how is it used.

[6 marks]

- (d) What is meant by the keyword **final**? Why would we define a field to be final? [3 marks]