

## A10.1-R4: INTRODUCTION TO OBJECT ORIENTED PROGRAMMING THROUGH JAVA

### NOTE:

1. There are **TWO PARTS** in this Module/Paper. **PART ONE** contains **FOUR** questions and **PART TWO** contains **FIVE** questions.
2. **PART ONE** is to be answered in the **TEAR-OFF ANSWER SHEET** only, attached to the question paper, as per the instructions contained therein. **PART ONE** is **NOT** to be answered in the answer book.
3. Maximum time allotted for **PART ONE** is **ONE HOUR**. Answer book for **PART TWO** will be supplied at the table when the answer sheet for **PART ONE** is returned. However, candidates, who complete **PART ONE** earlier than one hour, can collect the answer book for **PART TWO** immediately after handing over the answer sheet for **PART ONE**.

**TOTAL TIME: 3 HOURS**

**TOTAL MARKS: 100**  
**(PART ONE – 40; PART TWO – 60)**

### **PART ONE** **(Answer all the questions)**

1. **Each question below gives a multiple choice of answers. Choose the most appropriate one and enter in the “tear-off” answer sheet attached to the question paper, following instructions therein. (1x10)**
  - 1.1 In UML diagram of a class
    - A) state of object cannot be represented
    - B) state is irrelevant
    - C) state is represented as an attribute
    - D) state is represented as a result of an operation
  - 1.2 What is the default encoding for an OutputStreamWriter?
    - A) UTF8
    - B) Unicode
    - C) 8859\_1
    - D) The default is system-dependent.
  - 1.3 What will be the result of invoking the wait() method on an object without ensuring that the current thread holds the lock of the object?
    - A) The code will fail to compile.
    - B) Nothing special will happen.
    - C) An IllegalMonitorStateException will be thrown if the wait() method is called, while the current thread does not hold the lock of the object.
    - D) The thread will be blocked until it gains the lock of the object.
  - 1.4 Encapsulation in object-oriented modelling is useful as
    - A) it allows improving methods of an object independent of other parts of a system
    - B) it hides implementation details of methods
    - C) it allows easy designing
    - D) encapsulates attributes and operations of object

1.5 What will be the result of attempting to compile and run the following program?

```
public class MyClass {
    public static void main(String[] args) {
        String s = "hello";
        StringBuilder sb = new StringBuilder(s);
        sb.reverse();
        if (s == sb) {
            System.out.println("a");
        }
        if (s.equals(sb)) {
            System.out.println("b");
        }
        if (sb.equals(s)) {
            System.out.println("c");
        }
    }
}
```

- A) The code will fail to compile because the constructor of the String class is not called properly.
- B) The code will fail to compile because the expression (s == sb) is illegal.
- C) The code will fail to compile because the expression (s.equals(sb)) is illegal.
- D) The program will print c, when run.

1.6 What is the return type of the hashCode() method in the Object class?

- A) String
- B) integer
- C) long
- D) Object

1.7 In object-oriented design

- A) operations and methods are identical
- B) methods do not change values of attributes
- C) methods and constructor are same
- D) methods specify algorithms, whereas operations only state what is to be done

1.8 Given that the name of the class MyClass is specified correctly, which commands are syntactically valid.

- A) Java -Ddebug=true MyClass
- B) Java -ddebug=true MyClass
- C) Java -Ddebug="true" MyClass
- D) Java -D debug=true MyClass

1.9 Which of the following is not reserved keyword?

- A) public
- B) static
- C) void
- D) main

1.10 What will be the result of compiling and running the following program?

```
public class NewClass2 {
    public static void main(String[] args) {
        NewClass2 obj = new NewClass2(n);
    }
    static int i = 5;
    static int n;
    int j = 7;
    int k;
    public NewClass2(int m) {
        System.out.println(i + ", " + j + ", " + k + ", " + n + ", " + m);
    }
    {
        j = 70;
        n = 20;
    } // Instance Initializer Block
    static {
        i = 50;
    } // Static Initializer Block
}
```

- A) The code will fail to compile because the instance initializer block tries to assign a value to a static field.
- B) The code will fail to compile because the field k will be uninitialized when it is used.
- C) The code will compile and print 50, 70, 0, 20, 0 when run.
- D) The code will compile and print 50, 70, 0, 20, 20 when run.

2. **Each statement below is either TRUE or FALSE. Choose the most appropriate one and ENTER in the “tear-off” sheet attached to the question paper, following instructions therein. (1x10)**

- 2.1 Objects selected to model a system must have all attributes which are invariant during operations of a system.
- 2.2 Use cases are not beneficial for all interface requirements.
- 2.3 Java considers the variables *number* and *NUMbEr* to be identical.
- 2.4 An instance member belongs to an instance, not to the class as a whole.
- 2.5 If the references x and y denote two different objects, the expression x.equals(y) is always false.
- 2.6 Subclasses of the String class can be mutable.
- 2.7 A local variable in a block cannot be redeclared in another block even if the blocks are disjoint.
- 2.8 If an exception is thrown during the execution of the finalize() method of an eligible object, the exception is ignored and the object is destroyed.
- 2.9 A constructor can be declared private.
- 2.10 The import declarations must be the first statement after any package declaration in a source file.

3. Match words and phrases in column X with the closest related meaning/ word(s)/phrase(s) in column Y. Enter your selection in the “tear-off” answer sheet attached to the question paper, following instructions therein. (1x10)

X		Y	
3.1	One of the common things said about objects is that they have	A.	inherited
3.2	Java does not support literals in	B.	method overloading
3.3	Object serialization allows an object to be transformed into a sequence of bytes that can later be re-created (deserialized) into the original object. This mechanism is generally known as	C.	lang package
3.4	A class declaration, without the extends clause, implicitly extends	D.	widening conversion
3.5	Several method implementations may have the same name, as long as the method signatures differ. This is called	E.	final
3.6	If a package declaration is omitted in a compilation unit, the Java byte code for the declarations in the compilation unit will belong to an	F.	persistence
3.7	A class can be declared with the keyword abstract to indicate that it cannot be	G.	a loss of precision
3.8	A widening conversion from an int or a long value to a float value, or from a long value to a double value, may result in	H.	instantiated
3.9	Variable increment (++) and decrement (--) operators cannot be applied to a variable that is declared	I.	the object class
3.10	Constructors cannot be	J.	identity
		K.	parent class
		L.	unnamed package
		M.	binary notation

4. Each statement below has a blank space to fit one of the word(s) or phrase(s) in the list below. Enter your choice in the “tear-off” answer sheet attached to the question paper, following instructions therein. (1x10)

A.	Byte code	B.	package	C.	JAR (Java Archive)
D.	ASCII	E.	arrays	F.	import
G.	private	H.	polymorphic	I.	CASE tool
J.	Inheritance	K.	Interoperability	L.	heap
M.	Protected				

- 4.1 A \_\_\_\_\_ operation has same name but uses different methods depending on class.
- 4.2 \_\_\_\_\_ verifier confirms that all security and portability are valid and do not violate Java’s security restrictions.
- 4.3 Storage for objects is allocated in a designated part of the memory called the \_\_\_\_\_ which has a finite size.
- 4.4 \_\_\_\_\_ is defined as cooperation between object codes produced by different software vendors and written in different programming languages.
- 4.5 A(n) \_\_\_\_\_ declaration is not required if you always refer to a class with its fully qualified class name.
- 4.6 In Java, \_\_\_\_\_ are objects.
- 4.7 Once algorithms are finalized, we convert them into a program using OOPL or using \_\_\_\_\_.
- 4.8 A \_\_\_\_\_ in Java is an encapsulation mechanism that can be used to group related classes, interfaces, enums, and sub packages.
- 4.9 The \_\_\_\_\_ utility provides a convenient way of bundling and deploying Java programs.
- 4.10 \_\_\_\_\_ in object-oriented modelling can be used to generalize and to specialize classes.

**PART TWO**  
(Answer any **FOUR** questions)

- 5.**
- a) What is Garbage Collection? List the techniques to facilitate garbage collection.
  - b) Constructor declarations are very much like method declarations. However, some restrictions on constructors should be applied. Explain those restrictions. Write the general syntax of constructor.
  - c) What are user defined exceptions in Java? When are user defined exception handling used?  
**(4+6+5)**
- 6.**
- a) What is JDBC and how you connect Java with database? List the types of JDBC drivers.
  - b) Draw class model (diagram) for order processing. The classes should show their attributes, and their operations. Also show the relationships between classes.
  - c) What are the component and container classes? Explain how to create a borderless window. Explain the purposes of component's requestFocus method.  
**(4+6+5)**
- 7.**
- a) Write a Java program to find whether a given string is a palindrome or not.
  - b) What is the difference between an applet and an application? Why can't applet read or write to files? How can a particular HTML page be loaded within an applet?
  - c) What is Multithreading? Can you restart stopped thread in Java? How do you synchronize Java object?  
**(4+6+5)**
- 8.**
- a) What is modeling? What are the advantages of creating a model? What are the different views that are considered when building an object-oriented software system?
  - b) What is the use of inheritance? Why multiple inheritances are not possible in Java? What is markable interface in Java?
  - c) What is the significance of JVM? Why is the JVM platform dependent? Write the Difference between JVM and Java compiler.  
**(4+6+5)**
- 9.**
- a) List and explain briefly the concepts of object oriented programming.
  - b) What are the various components in sequence diagrams? What are the parts of a deployment diagram?
  - c) Explain the different access modifiers in Java. Why should double data type is preferable over float data type? What is importance of static variable?  
**(5+5+5)**