



EASTERN UNIVERSITY, SRI LANKA  
DEPARTMENT OF MATHEMATICS  
THIRD YEAR EXAMINATION IN SCIENCE - 2014/2015  
SECOND SEMESTER (OCT./NOV., 2018)  
OC 306 - FUNDAMENTAL OF JAVA PROGRAMMING

---

Answer all questions

Time allowed: Two Hours

---

- Q1. Java is a powerful and versatile programming language that produces software for multiple platforms.
- a. State what is meant by *Java Virtual Machine (JVM)* and whether it is platform independent. Justify your answer. [4%]
  - b. Briefly discuss *enum* type with suitable example. [3%]
  - c. State what are *wrapper classes*. Briefly explain its advantages. [4%]
  - e. State whether the following statements are true or false. Justify your answer. [14%]
    - i. *new* operator is used to create both primitive types and objects.
    - ii. *static* variable can not be modified.
    - iii. *final* can be used to prevent a method from being overloaded.
    - iv. A class can inherit exactly one interface.
    - v. *void Rect()* is the constructor of Rect Class.
    - vi. Method overriding can be performed within a single class.
    - vii. Object can be created for abstract class.

Q2. Repetition cause a section of the program to be repeated a certain number of times. The repetition continues while a condition is true.

a. Find the output of the following Java program: [5]

```
public class Test {  
    public static void main(String[] args) {  
        int i = 5;  
        int num;  
        while (i >= 1){  
            num = 1;  
            for (int j = 1; j <= i; j++){  
                System.out.print(num + "xxx");  
                num *= 2;  
            }  
            System.out.println();  
            i--;  
        }  
    }  
}
```

b. Answer the following, [6]

- i. Write a Java method called *swapDigits* to accept a two-digit integer and return the number formed by swapping its two digits. [ Example: input 12 will returned as 21 ]
- ii. Write a Java method called *leapYear* to accept the four-digit value of a given year and determine whether it is a leap year or not.

c. Explain the functionality of following functions. Find the output if the value of n

5. [4]

```
i. int fun1(int n) {  
    if(n == 1)  
        return 0;  
    else  
        return 1 + fun1(n/2);  
}
```

```

ii. void fun2(int n) {
    int i = 0;
    if (n > 1)
        fun2(n-1);
    for (i = 0; i < n; i++)
        System.out.print(" * " + " ");
}

```

d. Explain the concept *exception-handling* and briefly discuss the *try-catch* exception handling mechanism in Java. [5%]

e. Write Java programs to illustrate each of the following exceptions. [5%]

i. Arithmetic Exception

ii. ArrayIndexOutOfBoundsException

Q3. Object Oriented programming (OOP) is an approach to organise programs. OOP is a programming paradigm that uses abstraction to create models based on the real world environment.

a. What is *class* and *object* in Java? [4%]

b. Trace the following two programs Test1.java and Test2.java and write down the corresponding outputs. Explain the difference between the following two programs, which lead to different outputs. [4%]

<pre> //program Test1.java public class Test1{     int a,b;     Test1(int a, int b){         a=a;         b=b;     }     public static void main(String[] args){         Test1 obj = new Test1(1,2);         System.out.println(obj.a);         System.out.println(obj.b);     } } </pre>	<pre> //program Test2.java public class Test2{     int a,b;     Test2(int a, int b){         this.a=a;         this.b=b;     }     public static void main(String[] args){         Test2 obj = new Test2(1,2);         System.out.println(obj.a);         System.out.println(obj.b);     } } </pre>
---	---

- c. By using suitable examples, explain the following techniques in Java. [6]
- i. Constructor chaining, and
  - ii. Constructor overloading.
- d. Describe the differences between *static* and *instance* variable with the aid of suitable example. [5]
- e. Explain the use of keywords *extends*, *super*, and *abstract*. [6]

Q4. All object-oriented programming languages provide mechanisms that help to implement the object-oriented model.

- a. How do the *abstract classes* differ from *interfaces* in Java Programming language? [6]
- b. Assume that you have written a class whose attributes are encapsulated, but need to be accessed outside the class. [6]
  - i. Explain what type of attributes and methods to be included in the class.
  - ii. Demonstrate your answer by defining a class representing account with attributes *accountNumber* and *balance*.
- c. State what is meant by *multiple inheritance* and is it supported by Java. Justify your answer.
- d. A system is required for managing data processing in a University. Four different classes in the system are *Student*, *Course*, *Optional* and *Compulsory*. For each class, a method is required for attribute initialisation and another method to display summary. The summary of a *Student* instance contains the *name* of the student and his/her *registration number*. The summary of an *Optional* instance contains its *registration number* and the *number of credits*. The summary for a *Compulsory* instance is its *title* of the module, its *code* and the *number of credits*.

Implement the four classes in the following way:

- i. *Optional* and *Compulsory* as subclasses of a common abstract class *Course* which provides necessary interface to achieve polymorphism.
- ii. *Optional* and *Compulsory* classes must override the display method.
- iii. *Student* class as a concrete class.
- iv. Write down a main method to instantiate the necessary objects.