EASTERN UNIVERSITY, SRI LANKA

DEPARTMENT OF MATHEMATICS

THIRD YEAR SECOND SEMESTER (NOV. / DEC., 2017) - 2014/2015

OC 306 – FUNDAMENTAL OF JAVA PROGRAMMING

nswer All questions **Time allowed: Two Hours**

1) Java is a powerful and versatile programming language for developing software running on mobile devices, desktop computers, and servers.

a) Why Java is said to be **Platform Independent** Language? Justify your answer. [4%]

b) Explain the difference between each of the following pair using suitable examples:

(i) unary and binary operators, [2%]

(ii) relational and logical operators, [2%]

(iii) prefix and postfix operators, and [2%]

(iv) left-shift and right-shift bitwise operators. [2%]

c) Find the output of the following JAVA code fragment:

```
int a=2, b=16;
System.out.println("a is  " + a);
System.out.println("b is  " + b);
System.out.println("~a << 2 is  " + (~a));
System.out.println("a | b is  " + (a|b));
System.out.println("a & 2 is  " + (a & 2));
System.out.println("b & 1 is  " + (b & 1));
System.out.println("++a << 2 is  " + (++a << 2));
System.out.println("b-- >> 2 is  " + (b-- >> 2));
System.out.println("a is  " + a);
System.out.println("b is  " + b);
```

[8%]

d) Describe the difference between **syntax errors** and **semantic errors**. State three situations in which compilation errors are generated. [5%]

Q2) A control structure is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters.

a) State the keywords **break** and **continue** in Java. Show the output after the **continue** and **break** statement in the following loops. [5%]

```
int i=1,x=0;
   while(i<=5){
         x++;
         System.out.println("x = "+x);
         if(x<4)
            continue;
         System.out.println("i = "+i);
         i++;
         }
}
```

```
int i=1,x=0;
   while(i<=5){
         x++;
         System.out.println("x = "+x);
         If (x<4)
            break;
         System.out.println("i = "+i);
         i++;
         }
}
```

b) Discuss the difference between **Array** and **Array-List** in Java. [4%]

c) Write a program with the following procedure:

(i) Create a public class named **TestQ2** which includes the main method. [2%]

(ii) Declare and instantiate an integer array named **number** for 10 elements within the main method. [2%]

(iii) Using **Scanner** class, allow users to input 10 integers into the **number** array. [2%]

(iv) Create a static method called **findSum** which can receive the number array and return the sum of its 10 elements to the calling method. [3%]

(v) Invoke the **findSum** method with appropriate parameters and display the sum of the 10 element of the array. [2%]

d) What is an **exception**? Describe the **exception handling mechanism** in Java. [5%]

Q3) Methods can be used to define reusable code through which coding can be organised and simplified.

a) Why is the **main** method in Java declared as static? [3%]

b) Explain the difference between **method** and **constructor** in Java. [5%]

c) Answer the following:

(i) Using Java, write a method called **checkEven** that accepts an integer parameter and returns whether it is *even* or *odd*.

(ii) Using Java, write a method called **difference Of Digits** that accepts a two digit integer and returns the difference of the digits.

(iii) Using Java, write a method called **Larger** that accepts a two numbers and returns the largest number.

[3×3%]

d) By using suitable examples, explain the following techniques in Java:

(i) Constructor chaining, [2%]

(ii) Constructor overloading. [2%]

e) Discuss the modifiers **static** and **final** in Java. [4%]

Q4) Object Oriented Programming is a programming paradigm that uses abstraction to create models based on the real world environment.

a) Define the keywords **this** and **super** in Java. Write a program to emphasize the difference between **this** and **super** keywords. [5%]

b) Briefly explain the terms **abstraction, abstract class** and **abstract method**. [3%]

c) Explain the difference between **Class** and **Interface** in Java? [3%]

d) Design a class named **Fan** to represent a fan. The class contains:

(i) Three constants named **SLOW, MEDIUM**, and **FAST** with the values **1, 2**, and **3** to denote the fan as speed. [1.5%]

(ii) A private **int** data field named **speed** specifies the speed of the fan (the default value is **SLOW**). [0.5%]

(iii) A private **boolean** data field named **ON** specifies whether the fan is on or not (the default value is **false**). [0.5%]

(iv) A private **double** data field named **radius** specifies the radius of the fan (the default value is **5**). [0.5%]

(v) A string data field named **color** specifies the colour of the fan (the default value is **blue**). [0.5%]

(vi) The **get** and **set** methods for all four data fields. [2%]

(vii) A constructor that creates a default Fan without accepting any parameter(s). [0.5%]

(viii) A method named **display()** that returns a string description for the Fan. If the fan is on, the method returns the fan speed, colour, and radius in one combined string. If the fan is not on, the method returns the fan colour and radius along with the string "fan is off" in one combined string. [2%]

e) Write a test program that creates two **Fan** objects. Assign maximum speed, radius **10**, colour **yellow**, and turn it ON to the first object. Assign medium speed, radius **5**, colour **blue**, and turn it OFF to the second object. Display the objects by invoking their **display** method. [3%]