



**UNIVERSITI TUN HUSSEIN ONN MALAYSIA**

**FINAL EXAMINATION  
SEMESTER II  
SESSION 2011/2012**

**COURSE NAME** : JAVA PROGRAMMING  
**COURSE CODE** : BIT 3383/BIT 33803  
**PROGRAMME** : BACHELOR OF INFORMATION  
TECHNOLOGY  
**EXAMINATION DATE** : JUNE 2012  
**DURATION** : 2 HOURS 30 MINUTES  
**INSTRUCTION** : ANSWER ALL QUESTIONS.

**THIS QUESTION PAPER CONSISTS OF SIX (6) PAGES**

**SECTION A**

**Instruction:** State whether each of the following statement is **TRUE** or **FALSE**. **ONE (1) mark** for each correct answer.

- Q1** Comments in Java programming must end with semicolons.
- Q2** Java is not a case-sensitive language. Therefore, it regards uppercase letters as being the same character as their lowercase equivalents.
- Q3** Class headers and method headers can end either with a semicolon or braces.
- Q4** A `.java` file may contain many classes, but may only have one public class.
- Q5** Every Java application program must have a method named `main`.
- Q6** Objects of Scanner class can be used to read input from the keyboard.
- Q7** A superclass constructor always executes before subclass constructor.
- Q8** We can write a `super` statement that calls a superclass constructor, but only in the subclass's constructor.
- Q9** A subclass may have a method with the same signature as a superclass method. In such a case, the subclass method overloads the superclass method.
- Q10** A class is the blueprint for an object. It specifies the attributes and methods that a particular type of object has.
- Q11** An enumerated data type consists of a set of predefined values. We can use the data type to create variables that can hold only the values that belong to the enumerated data type.
- Q12** Part of object-oriented design process in Java programming is identifying collaborations between classes.
- Q13** An array can hold multiple values of different data type simultaneously.

- Q14** We may create arrays of objects that are instances of classes that we have written.
- Q15** An exception is an object that is generated as the result of an error or an unexpected event.
- Q16** We use Java Foundation Classes to create graphical user interface for our application. Within it, we can use the Abstract Windowing Toolkit or Swing classes.
- Q17** The `JOptionPane` class allows us to display a dialog box for displaying message or requesting input.
- Q18** The main method that Java uses as the starting point of an application cannot be written directly into a GUI class.
- Q19** By writing a class that inherits from `JPanel`, we can create a custom panel component that can hold other components and their related code.
- Q20** The `JList` component can operate in single selection mode and single interval selection mode only.

(20 marks)

**SECTION B****Instruction: Answer ALL questions.****Q21 Determine the output for Figure Q21. Show the steps to produce the output.**

```

/*This is MyClass class which contains testMe method*/
public class MyClass
{
    public static void testMe(int[] array)
    {
        int startScan, index, minIndex, minValue;

        for (startScan = 0; startScan < (array.length-1); startScan++)
        {
            minIndex = startScan;
            minValue = array[startScan];
            for(index = startScan + 1; index < array.length; index++)
            {
                if (array[index] < minValue)
                {
                    minValue = array[index];
                    minIndex = index;
                }
            }
            array[minIndex] = array[startScan];
            array[startScan] = minValue;
        }
    }
}

/* This program demonstrates the testMe method in the MyClass class. */
public class Demo
{
    public static void main(String[] arg)
    {
        int[] values = {35, 17, 23, 12, 18, 48, 19, 21, 27, 11};

        System.out.println("Line 1 values are:");
        for (int index = 0; index < values.length; index++)
            System.out.print(values[index] + " ");
        System.out.println();

        MyClass.testMe(values);

        System.out.println("The output values are:");
        for (int index = 0; index < values.length; index++)
            System.out.print(values[index] + " ");
    }
}

```

**Figure Q21****(20 marks)**

**Q22** Identify and fix **FIFTEEN (15)** errors in **Figure Q32**. In your answer, indicate the explanation of the errors, and their corresponding corrections such as in the following example:

**Example:**

Line 1: wrong spelling of import , import java.util.Scanner;

Correction: import java.util.Scanner;

Note that a sample of output screen that we aim to achieve for **Figure Q32** is given as in the following:

**Sample output:**

For how many days do you have sales figures? 5

Enter the sales for day 1: 687.59

Enter the sales for day 2: 563.22

Enter the sales for day 3: 896.35

Enter the sales for day 4: 743.29

Enter the sales for day 5: 926.72

The total sales are           \$3,817.17

```

1. import java.util.Scanner;
2. import java.text.DecimalFormat;
3. /**
4.  * This program calculates a running total.
5.  public class TotalSales
6.
7.  public static main(String[] args)
8.  { int days;
9.    int sales;
10.     int totalSales;
11.
12.     Scanner keyboard = new Scanner(System.out);
13.     DecimalFormat dollar = new DecimalFormat("#,##0.00");
14.     System.out.print("For how many days do you have "
15.                      "sales figures? ");
16.     days = keyboard.nextDouble();
17.     totalSales = 0.0;
18.
19.     for (count = 1; count <= days; count++);
20.     System.out.print("Enter the sales for day "
21.                      + count +:;
22.     sales = keyboard.nextLine();
23.     totalSales += sales;    }
24.
25.     System.out("The total sales are $"
26.               + format(totalSales));  }}

```

**Figure Q22**

(15 marks)

**SECTION C**

**Instruction:** Answer **ALL** questions.

**Q23** Write a program that acts as a sentence analyzer. A user may enter a sentence and the program will determine number of letters, digits, and whitespace characters in the sentence. A sample output is given as follows:

**Sample output**

Enter a string: 27 red books

The string contains 8 letters, 2 digits, and 2 whitespace characters.

(20 marks)

**Q24** Write a class that will use a StringTokenizer object. Its constructor accepts a string containing a date in the form MONTH/DAY/YEAR. It extracts the month, day, and year and stores these values in the month, day, and year fields. The methods getMonth, getDay, and getYear can then be used to retrieve the values. Sample output upon calling the methods in the class is given as follows.

**Sample output**

The date is: 10/23/2011

The month is 10

The day is 23

The year is 2011

(25 marks)