Logic Building Assignment : 76

1. Java program to add two time class objects.

```
import java.lang.*;
import java.io.*;
import java.util.*;

class Time
{
    public int hr;
    public int min;
    public int sec;

    public Time(int value1, int value2, int value3)
    {
        hr = value1;
        min = value2;
        sec = value3;
    }

    public static Time AddTime(Time op1, Time op2)
    {
        Time result = new Time(0,0,0);

        result.hr = op1.hr + op2.hr;
        result.min = op1.min + op2.min;
        result.sec = op1.sec + op2.sec;

        result.min = result.min + (result.sec / 60);
        result.sec = result.sec % 60;

        result.hr = result.hr + (result.min / 60);
        result.min = result.min % 60;

        return result;
    }
}
```

2. Java application which creates customised Linked list of student

```
import java.lang.*;
import java.io.*;
import java.util.*;

// Class which represents the node from linked list
class Node
{
    // Charcteristics
    public int rno;          // Roll number
    public int marks;        // Marks of student
    public Node next;        // Next reference
    public String name;      // Name of student

    // Behaviours
    {
        next = null;        //  Default value
    }

    public Node()
    {
        rno = 0;
        name = null;
        marks = 0;
    }

    public Node(int rno,String name,int marks)
    {
        this.rno = rno;
        this.name = name;
        this.marks = marks;
    }
} // End of node


// Class which creates and manage the linked list
class Student
{
    // Characteristics
    public Node head;

    // Behaviours
```

```
public Student()
{
    head = null;
}

public boolean insert(int no,String name, int marks)
{
    Node newn = new Node(no,name,marks);

    if(head == null)
    {
        head = newn;
    }
    else
    {
        newn.next = head;
        head = newn;
    }

    return true;
}

public void search(int no)
{
    Node temp = head;

    while(temp != null)
    {
        if(temp.rno == no)
        {
            System.out.print(temp.rno);
            System.out.print(" "+temp.name);
            System.out.print(" "+temp.marks);
            System.out.println();

            break;
        }
        temp = temp.next;
    }

    if(temp == null)
    {
        System.out.println("There is no sush student");
    }
    }

    public void search(String str)
    {
        Node temp = head;

        while(temp != null)
        {
            if(str.equals(temp.name))
            {
                System.out.print(temp.rno);
                System.out.print(" "+temp.name);
                System.out.print(" "+temp.marks);
                System.out.println();

                break;
            }
            temp = temp.next;
        }
        if(temp == null)
        {
            System.out.println("There is no sush student");
        }
    }

    public void delete(int no)
    {
        Node temp = head;
        Node deltenode = null;

        if(temp.rno == no)  // For first node
        {
            head = head.next;
        }

        // Fore remainonig nodes
        while(temp.next != null)
        {
            if(temp.next.rno == no)
            {
                          System.out.println("Information of node that you want to
delete : ");
                System.out.print(temp.next.rno);
                System.out.print(" "+temp.next.name);
                System.out.print(" "+temp.next.marks);
                System.out.println();

                break;
            }
            temp = temp.next;
        }

        if(temp.next == null)
        {
            System.out.println("There is no sush student");
            return;
        }

        System.out.println("Are you sure to delete the node 1/0");
        Scanner sobj = new Scanner(System.in);
        int option = sobj.nextInt();
        if(option == 0)
        {
            return;
        }
        else
        {
            deltenode = temp.next;
            temp.next = deltenode.next;

            System.out.println("Member deleted successfully");
        }
    }

    public void Update(int no)
    {
        Node temp = head;

        while(temp != null)
        {
            if(temp.rno == no)
            {
                System.out.println("Old information is : ");
                System.out.print(temp.rno);
                System.out.print(" "+temp.name);
                System.out.print(" "+temp.marks);
                System.out.println();
                break;
            }
            temp = temp.next;
        }

        if(temp == null)
        {
            System.out.println("There is no sush student");
            return;
        }

        Scanner sobj = new Scanner(System.in);

        System.out.println("Enter new roll number");
        temp.rno = sobj.nextInt();

        System.out.println("Enter new name");
        temp.name = sobj.next();

        System.out.println("Enter new marks");
        temp.marks = sobj.nextInt();

        System.out.println("Update succesfull..");
    }


    public void Display()
    {
        Node temp = head;

        while(temp != null)
        {
            System.out.print(temp.rno);
            System.out.print(" "+temp.name);
            System.out.print(" "+temp.marks);
            System.out.println();

            temp = temp.next;
        }
    }

    public void MaximumMarks()
    {
        if(head == null)
```

```
        {
            return;
        }

        Node temp = head;
        Node maxref = null;
        int max = 0;

        while(temp != null)
        {
            if(temp.marks > max)
            {
                max = temp.marks;
                maxref = temp;
            }

            temp = temp.next;
        }

        if(maxref != null)
        {
            System.out.println("Information of student with max marks :");
            System.out.print(maxref.rno);
            System.out.print(" "+maxref.name);
            System.out.print(" "+maxref.marks);
            System.out.println();
        }
    }
}

// Entry point class which contains main
class DD
{
    public static void main(String ar[])
    {
        Student sobj1 = new Student();
        Student sobj2 = new Student();
        Student sobj3 = new Student();
        Student sobj4 = new Student();

        sobj1.insert(11,"ABC",200);
        sobj1.insert(21,"PQR",300);
        sobj1.insert(51,"XYZ",400);
        sobj1.insert(101,"MNP",500);
        sobj1.insert(121,"BJP",600);
        sobj1.insert(151,"PAPPU",0);
        sobj1.Display();

        System.out.println();

        sobj1.search(101);
        sobj1.search(100001);
        System.out.println();
        sobj1.search("MNP");
        sobj1.search("PPP");
        System.out.println();

        sobj1.MaximumMarks();

        System.out.println();
        sobj1.delete(51);
        sobj1.Display();

    }
}
```

3. Java program to count frequency of each character from string.

```
class Demo
{
   static void characterCount(String inputString)
   {
       //Creating a HashMap containing char as a key and occurrences as a value

       HashMap<Character, Integer> charCountMap = new HashMap<Character, Integer>();

      //Converting given string to char array

      char[] strArray = inputString.toCharArray();

      //checking each char of strArray

      for (char c : strArray)
      {
         if(charCountMap.containsKey(c))
         {
              //If char is present in charCountMap, incrementing it's count by 1

            charCountMap.put(c, charCountMap.get(c)+1);
         }
         else
         {
            //If char is not present in charCountMap,
            //putting this char to charCountMap with 1 as it's value

            charCountMap.put(c, 1);
         }
      }

      //Printing the charCountMap

      System.out.println(charCountMap);
   }

   public static void main(String[] args)
   {
      characterCount("Java J2EE Java JSP J2EE");
```

```
        characterCount("All Is Well");

        characterCount("Done And Gone");
    }
}
```

4. Java program to check whether the first string is rotation of second or not.

```
public class MainClass
{
    public static void main(String[] args)
    {
        String s1 = "JavaJ2eeStrutsHibernate";

        String s2 = "StrutsHibernateJavaJ2ee";

        //Step 1

        if(s1.length() != s2.length())
        {
            System.out.println("s2 is not rotated version of s1");
        }
        else
        {
            //Step 2

            String s3 = s1 + s1;

            //Step 3

            if(s3.contains(s2))
            {
                System.out.println("s2 is a rotated version of s1");
            }
            else
            {
                System.out.println("s2 is not rotated version of s1");
            }
        }
    }
}
```

5. Java program to print matrix in spiral format

```
public class spyral
{
    public static void printSpiralOrder(int mat[][])
    {
        int top = 0, bottom = mat.length - 1;
        int left = 0, right = mat[0].length - 1;

        while (true)
        {
            if (left > right)
                break;

            // print top row
            for (int i = left; i <= right; i++)
                System.out.println(mat[top][i]);
            top++;

            if (top > bottom)
                break;

            // print right column
            for (int i = top; i <= bottom; i++)
                System.out.println(mat[i][right]);
            right--;

            if (left > right)
                break;

            // print bottom row
            for (int i = right; i >= left; i--)
                System.out.println(mat[bottom][i]);
            bottom--;

            if (top > bottom)
                break;

            // print left column
            for (int i = bottom; i >= top; i--)
                System.out.println(mat[i][left]);
            left++;
        }
    }
    public static void main(String[] args)
    {
        int arr[][]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};
        printSpiralOrder(arr);
    }
}
```