

Journal -3

1. Write a Java Program to implement Thread by Extending Thread Class.

Program:-

```
class th1 extends Thread
{
    public void run()
    {
        for(int i=1;i<10;i++)
        {
            System.out.println("Th1= "+i);
        }
    }
}

class th2 extends Thread
{
    public void run()
    {
        for(int i=1;i<10;i++)
        {
            System.out.println("Th2= "+i);
        }
    }
}

public class ques1 {

    public static void main(String[] args)
    {
```

```

th1 a1 = new th1();

th2 a2 =new th2();

a1.run();

a2.run();

System.out.println("exit main....");

}

}

```

Output:-

```

[Running] cd "d:\JAVA ASS-2\java_ass_3\" && javac ques1.java && java ques1
Th1= 1
Th1= 2
Th1= 3
Th1= 4
Th1= 5
Th1= 6
Th1= 7
Th1= 8
Th1= 9
Th2= 1
Th2= 2
Th2= 3
Th2= 4
Th2= 5
Th2= 6
Th2= 7
Th2= 8
Th2= 9
exit main....

```

2. Write a Java Program to Implement Thread by Implement Runnable Method

```

public class ques2 implements Runnable {
    public void run()
    {
        System.out.println("Thread Is Running...");
    }
    public static void main(String[] args)
    {
        ques2 a1=new ques2();
        Thread th1= new Thread(a1);

        th1.start();
    }
}

```

Output:-

```
[Running] cd "d:\JAVA ASS-2\java_ass_3\" && javac ques2.java && java ques2
Thread Is Running...
```

3. Write a Java Program to Implement Thread Priority.

```
public class ques3 extends Thread
{
    public void run()
    {
        System.out.println("Running Thread is: "+Thread.currentThread().getName());
        System.out.println("RunningThreadpriority:"+Thread.currentThread().getPriority());
    }
    public static void main(String[] args)
    {
        ques3 m1 =new ques3();
        ques3 m2 = new ques3();
        m1.setPriority(Thread.MIN_PRIORITY);
        m2.setPriority(Thread.MAX_PRIORITY);
        m1.start();
        m2.start();
    }
}
```

Output:-

```
Running Thread is: Thread-1
Running Thread is: Thread-0
Running Thread priority:10
Running Thread priority:1
```

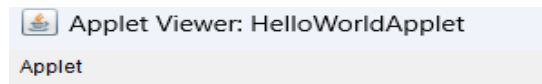
4. Write a Java Program to print "Hello World" in Applet.

Programm:-

```
import java.applet.*;
import java.awt.*;
/*<applet code="HelloWorldApplet" width="500" height="500">
    </applet>*/
public class HelloWorldApplet extends Applet
{
    public void paint(Graphics g)
    {
```

```
g.drawString("Hello World", 100, 100);  
}  
}
```

OutPut:-



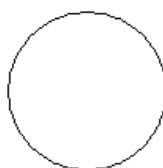
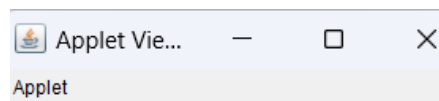
Hello World

5. Write a Java Program to Draw Circle in Applet

Programs:-

```
import java.applet.Applet;  
import java.awt.Graphics;  
/*<applet code="CircleApplet.class" width="300" height="300">  
</applet>*/  
public class CircleApplet extends Applet {  
    public void paint(Graphics g) {  
  
        int centerX = getWidth() / 2;  
        int centerY = getHeight() / 2;  
        int radius = 50;  
        g.drawOval(centerX - radius, centerY - radius, 2 * radius, 2 * radius);  
    }  
}
```

Output:-

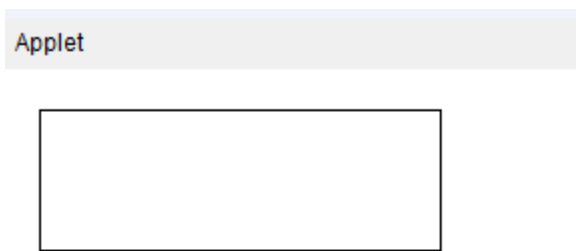


6. Write a Java Program to Draw Rectangle in Applet

```
import java.applet.Applet;
import java.awt.Graphics;
/*<applet code="RectangleApplet.class" width="300" height="300">
  </applet>*/

public class RectangleApplet extends Applet
{
    public void paint(Graphics g)
    {
        g.drawRect(20,20,200,70);
    }
}
```

Output:-



7. Write a Java Program to Draw Triangle in Applet

Program:-

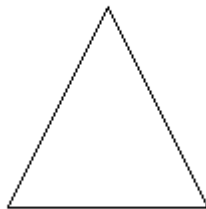
```
import java.applet.Applet;
import java.awt.Graphics;

public class TriangleApplet extends Applet
{
    public void paint(Graphics g)
    {
        int[] xPoints = {100, 150, 200};
        int[] yPoints = {150, 50, 150};
        int numPoints = 3;
        g.drawPolygon(xPoints, yPoints, numPoints);
    }
}
```

```
/*<applet code="TriangleApplet.class" width="300" height="200">
</applet>*/
```

Output:-

Applet



Applet started.

8. Write a Java Program to Draw Pentagon in Applet

Program:-

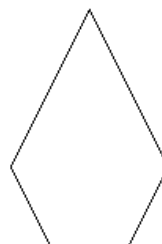
```
import java.applet.Applet;
import java.awt.Graphics;
public class PentagonApplet extends Applet
{
    public void paint(Graphics g)
    {
        int[] xPoints = {100, 150, 200, 175, 125};
        int[] yPoints = {150, 50, 150, 200, 200};
        int numPoints = 5;
        g.drawPolygon(xPoints, yPoints, numPoints);
    }
}
```

```
/*<applet code="PentagonApplet.class" width="300" height="250">
```

```
</applet>*/
```

Applet

OutPut:-



9. Write a Java Program to Draw Smile Face in Applet

Programs:=

```
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class SmileFaceApplet extends Applet
{
    public void paint(Graphics g)
    {
        setBackground(Color.white);
        g.setColor(Color.yellow);
        g.fillOval(50, 50, 200, 200);
        g.setColor(Color.black);
        g.fillOval(100, 100, 30, 30);
        g.fillOval(180, 100, 30, 30);
        g.drawArc(100, 120, 110, 100, 180, 180);
    }
}

/*<applet code="SmileFaceApplet.class" width="300" height="300">
</applet>*/
```

Output:-



10. Write a java Program to implement String buffer all methods.

Program:-

```
public class StrBuffer
{
    public static void main(String[] args)
    {
        StringBuffer str= new StringBuffer("Hello ");
        str.append("User");
        System.out.println("After append:-"+str);
        str.insert(5, "Java");
        System.out.println("after Insert"+str);
        str.delete(5, 10);
        System.out.println("After delete:- " + str);
        str.reverse();
        System.out.println("After reverse:- " + str);
        int length = str.length();
        System.out.println("Length of StringBuffer:- " + length);
        int capacity = str.capacity();
        System.out.println("Capacity of StringBuffer: -" + capacity);
        str.setLength(3);
        System.out.println("After setLength:- " + str);
        str.replace(0, 3, "Dear");
        System.out.println("After replace:- " + str);
        str.ensureCapacity(30);
        System.out.println("Capacity after ensureCapacity:- " + str.capacity());
        str.trimToSize();
        System.out.println("Capacity after trimToSize:- " + str.capacity());
        String s1 = str.toString();
        System.out.println("String representation:- " + s1);
    }
}
```

OutPut:-

```
[Running] cd "d:\JAVA ASS-2\java_ass_3\" && javac StrBuffer.java && java StrBuffer
After append:-Hello User
after InsertHelloJava User
After delete:- HelloUser
After reverse:- resUolleH
Length of StringBuffer:- 9
Capacity of StringBuffer: -22
After setLength:- res
After replace:- Dear
Capacity after ensureCapacity:- 46
Capacity after trimToSize:- 4
String representation:- Dear
```


11. Write a Java Program to implement Singly Link List

Program:-

```
// Java program to implement
// a Singly Linked List
public class LinkedList
{
    Node head; // head of list

    // Linked list Node.
    // This inner class is made static
    // so that main() can access it
    static class Node
    {
        int data;
        Node next;

        // Constructor
        Node(int d)
        {
            data = d;
            next = null;
        }
    }

    // Method to insert a new node
    public static LinkedList insert(LinkedList list, int data)
    {
        // Create a new node with given data
        Node new_node = new Node(data);

        // If the Linked List is empty,
        // then make the new node as head
        if (list.head == null) {
            list.head = new_node;
        }
        else {
            // Else traverse till the last node
            // and insert the new_node there
            Node last = list.head;
            while (last.next != null) {
                last = last.next;
            }

            // Insert the new_node at last node
            last.next = new_node;
        }

        // Return the list by head
        return list;
    }
}
```

```

// Method to print the LinkedList.
public static void printList(LinkedList list)
{
    Node currNode = list.head;

    System.out.print("LinkedList: ");

    // Traverse through the LinkedList
    while (currNode != null) {
        // Print the data at current node
        System.out.print(currNode.data + " ");

        // Go to next node
        currNode = currNode.next;
    }
}

// Driver code
public static void main(String[] args)
{
    /* Start with the empty list. */
    LinkedList list = new LinkedList();

    //
    // *****INSERTION*****
    //

    // Insert the values
    list = insert(list, 1);
    list = insert(list, 2);
    list = insert(list, 3);
    list = insert(list, 4);
    list = insert(list, 5);
    list = insert(list, 6);
    list = insert(list, 7);
    list = insert(list, 8);

    // Print the LinkedList
    printList(list);
}
}

```

OutPut:-

```
LinkedList: 1 2 3 4 5 6 7 8
```