Roll Number:

The LNM Institute of Information Technology
End Semester Exam (OOPs (using Java)

Max Marks: 40 **Max Time: 3 Hr.**

PART- A

1. **Correct the error if any and find the output of the following program segments.**
2. **Show workout in the answer script.**
3. **Marks are mentioned in the first row and right corner of each question.**

Q1 **[3]**

```
import java.util.*;
  class Output {
     public static void main(String args[]) {
        TreeSet t = new TreeSet();
        t.add("3");
        t.add("9");
        t.add("1");
        t.add("4");
        t.add("8");
        System.out.println(t);
     }
  }
```

Q2. Consider the following code segment: **[3]**

```
Picture picture = new Picture();
picture.add(new Line(100, 100, 200, 50));
picture.add(new Line(200, 50, 300, 100));
Picture box = new Picture();
box.add(new Line(100, 100, 100, 300));
box.add(new Line(100, 300, 300, 300));
box.add(new Line(300, 300, 300, 100));
box.add(new Line(300, 100, 100, 100));
```

Which shape will be displayed when pictures.draw(g) is called from an appropriate paint method within the graphics context g.

Q3. **[4]**

***TreeNode contains (value, left node, right node)**

```
TreeNode node6 = new TreeNode("6", null, null);
TreeNode node5 = new TreeNode("5", null, null);
TreeNode node4 = new TreeNode("4", null, null);
TreeNode node3 = new TreeNode("3", node5, node6);
TreeNode node2 = new TreeNode("2", null, node4);
TreeNode node1 = new TreeNode("1", node2, node3);
TreeNode root = node1;

Object[] arr = new Object[8];
toArray(root, 1, arr);

for (int i = 0; i < arr.length; i++)
   System.out.println(arr[i] + " ");
```

The method toArray is defined as follows:

```
private void toArray(TreeNode root, itn i, Object[] arr)
{
   if (root != null)
   {
      arr[i] = root.getValue();
      toArray(root.getLeft(), 2*i, arr);
      toArray(root.getRight(), 2*i + 1, arr);
   }
}
```

Q4. **[4]**

```
import java.awt.*;
import java.awt.event.*;
public class AWTButtons extends Frame {
  private TextField tfCount;
  private int count = 0;
  public AWTButtons () {
    setLayout(new FlowLayout());
    add(new Label("Counter"));
    tfCount = new TextField("0", 10);
    tfCount.setEditable(false);
    add(tfCount);
     Button btnCountUp = new Button("Count Up");
    add(btnCountUp);
    btnCountUp.addActionListener(new ActionListener() {
       public void actionPerformed(ActionEvent e) {
         ++count;
         tfCount.setText(count + "");
       }
    });
    setTitle("AWT Counter");
    setSize(400, 100);
    setVisible(true);
  }

  /** The entry main method */
  public static void main(String[] args) {
    new AWTButtons();
  }}
```

Q5. **[3]**

```
class s1 implements Runnable
{
   int x = 0, y = 0;
   int addX() {x++; return x;}
   int addY() {y++; return y;}
   public void run() {
   for(int i = 0; i < 10; i++)
      System.out.println(addX() + " " + addY());
}
   public static void main(String args[])
   {
      s1 run1 = new s1();
      s1 run2 = new s1();
      Thread t1 = new Thread(run1);
      Thread t2 = new Thread(run2);
      t1.start();
      t2.start();
   }
}
```

Q6. **[3]**

```
import java.util.*;
   class Collection_iterators {
      public static void main(String args[]) {
         LinkedList list = new LinkedList();
         list.add(new Integer(2));
         list.add(new Integer(8));
         list.add(new Integer(5));
         list.add(new Integer(1));
         list.remove(2);
         list.addFirst(new Integer(7));
         Iterator i = list.iterator();
         while(i.hasNext())
                System.out.print(i.next() + " ");
      }
   }
```

Q7. **[4]**

```
public class Circle
{
    private int xCenter, yCenter, radius;
    public Circle(int x, int y, int r)
    {
        xCenter = x;
        yCenter = y;
        radius = r;
    }
    public void moveTo(int x, int y)
    {
        xCenter = x;
        yCenter = y;
    }
    // Draw this circle in the graphics context g
    public void draw(Graphics g) {    < code to display circle >    }
}
```

Suppose the following code is added to a method that repaints a window within a graphics context g:

```
for (int x = 10; x <= 30; x += 10)
{
    Circle circle = new Circle(x + 100, 100, x);
    circle.draw(g);
}
```

The origin of the coordinate system is in the upper left corner of the window with the y-axis pointing down. Which image will be displayed if the above code is executed?

PART-B

Each question carry 8 marks

Q1. In computing, the producer–consumer's problem (also known as the bounded-buffer problem) is a classic example of a multi-process synchronization problem. The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer used as a queue. The producer's job is to generate a piece of data, put it into the buffer and start again. At the same time, the consumer is consuming the data (i.e., removing it from the buffer) one piece at a time. The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer.

Suggest and write monitor code in Java to state and resolve the above issue using Multithreading.

Q2. Design skeleton of a Vehicle Management System. It may involve four modules, Bus Management, Route Management, Employee Management and Passenger Management.

Implement only the Bus management module using **Swing** and **JDBC** for the following functionality:

1. User can add a new bus details to the database.
2. Bus details can be removed from the database if older than six years.