School of Computer Science Engineering and Technology

Course- BTech
Course Code- **CSET109**
Year- First
Type- Core
Course Name- **Object Oriented Programming Using Java**
Semester- Even Batch- BTech 2nd Semester

Tutorial-9

Tutorial No.	Name	CO1	CO2	CO3
9	**Multithreading and File handling**	✓	✓	**--**

Objective: The main objective of this tutorial is to learn about the multithreading and file handling concepts of Java language.

9.1 What will be the output of the following program?

```
class MyThread extends Thread {
   public static void main(String[] args) {
      MyThread t = new MyThread();
      t.run();
   }
   public void run() {
      for (int i = 1; i < 7; ++i) {
         System.out.print(i + "..");
      }
   }
}
```

9.2 What will be the output of the following program?

```
public class SpiderMan 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 < 5; i++)
         System.out.print(addX() + " " + addY() + ", ");
   }
   public static void main(String args[]) {
      SpiderMan run1 = new SpiderMan();
      SpiderMan run2 = new SpiderMan();
      Thread t1 = new Thread(run1);
      Thread t2 = new Thread(run2);
      t1.run();
      t2.run();
   }
}
```

9.3 What will be the output of the following Java program?

```
   public class Digital extends Thread {
```

```
    public void run() {
        System.out.println("Now you are here");
    }

    public static void main(String a[]) {
        Thread run = new Thread(new Digital());
        System.out.print("1. ");
        run.start();
        System.out.print("2. ");
        run.start();
    }
}
```

9.4 What will be the output of the following program?

```
import java.io.*;
    public class filesinputoutput
    {
          public static void main(String[] args)
        {
            String obj  = "abc";
          byte b[] = obj.getBytes();
          ByteArrayInputStream obj1 = new ByteArrayInputStream(b);
          for (int i = 0; i < 2; ++ i)
          {
             int c;
             while((c = obj1.read()) != -1)
             {
                    if(i == 0)
                {
                        System.out.print(Character.toUpperCase((char)c));
                   obj2.write(1);
                    }
             }
             System.out.print(obj2);
          }
        }
    }
```

9.5 What will be the output of the program?

```
public class TestRunMethod extends Thread
{
    public static void main(String[] args)
    {
        TestRunMethod t = new TestRunMethod();
        t.start();
        t.start();
    }
    public void run()
    {
```

```
        System.out.println("test");
    }
}
```

9.6 What will be the output of this program?

```
public class ThreadTest implements Runnable {
    public static void main(String[] args) {
        Thread thread = new Thread(new ThreadTest());
        thread.run(); // LINE A
        try {
            thread.join();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public void run() {
        for (int i = 2; i <= 10; i = i + 2) {
            System.out.print(i);
        }
    }
}
```

9.7 What will be the output of the following program?

```
public class PriorityDemo {
    Priority t1, t2, t3;
    public PriorityDemo() {
        t1 = new Priority();
        t1.run();
        t2 = new Priority();
        t2.run();
        t3 = new Priority();
        t3.start();
    }
    public static void main(String args[]) {
        new PriorityDemo();
    }
}
class Priority extends Thread implements Runnable {
    int sleep;
    static int prio = 3;
    public Priority() {
        sleep += 100;
        prio++;
        setPriority(prio);
    }
    public void run() {
        try {
            Thread.sleep(sleep);
            System.out.print(getName() + " Priority = " + getPriority() + ", ");
        } catch (InterruptedException e) {
```

```
            System.out.println(e.getMessage());
        }
    }
}
```

9.8 What will be the output of this program?

```
import java.io.*;
    class Chararrayinput
    {
        public static void main(String[] args)
        {
            String obj  = "abcdef";
            int length = obj.length();
            char c[] = new char[length];
            obj.getChars(0, length, c, 0);
            CharArrayReader input1 = new CharArrayReader(c);
            CharArrayReader input2 = new CharArrayReader(c, 0, 3);
            int i;
            try
            {
                    while((i = input2.read()) != -1)
                {
                    System.out.print((char)i);
                }
                }
            catch (IOException e)
            {
                e.printStackTrace();
                }
            }
    }
```

9.9 What will be output of the following program?

```
public class Cruiser {
    private int a = 0;
    public void meth() {
        Runnable r = new LittleCruiser();
        new Thread(r).start();
        new Thread(r).start();
    }
    public static void main(String arg[]) {
        Cruiser c = new Cruiser();
        c.meth();
    }
    public class LittleCruiser implements Runnable {
        public void run() {
            int current = 0;
            for (int i = 0; i < 4; i++) {
                current = a;
                System.out.print(current + ", ");
```

```
            a = current + 2;
        }
    }
  }
}
```

9.10 What will be the output of the following program?

```
class NThread implements Runnable {
   String str;
   Thread t;
   NThread(String nameofthread) {
      str = nameofthread;
      t = new Thread(this, str);
      t.run();
   }
   public void run() {
      try {
         for (int i = 5; i > 3; i--) {
            System.out.print(str + " : " + i + ", ");
            Thread.sleep(100);
         }
      } catch (InterruptedException e) {    System.out.println(str + "Interrupted");    }
      System.out.println(str + " exiting");
   }
}
class MultiThreadDemo {
   public static void main(String[] args) {
      new NThread("FIRST");
      new NThread("SECOND");
      new NThread("THIRD");
      try {
         Thread.sleep(100);
      } catch (InterruptedException e) {    System.out.println("Main Thread Interrupted");    }
      System.out.println("Main Thread Exiting");
   }
}
```