SANGOLA COLLEGE, SANGOLA
Class-B.Sc(ECS)-II, SEM-IV 2024-25
Practical Assignments
Sub- Core Java

Assignment No- 2

1. Write a program which handles multiple exceptions(Single try and multiple catch blocks)

```
class Mul_Catch
{
   public static void main(String arg[])
   {
      try
      {
         int a[] = new int[3];
         System.out.print("Element at index 5 is: " + a[5]);

         int c = 7 / 0;
      }
      catch(ArithmeticException e)
      {
         System.out.print("Catch: " + e);
      }
      catch(ArrayIndexOutOfBoundsException e)
      {
         System.out.print("Catch: " + e);
      }
      catch(Exception e)
      {
         System.out.print("Catch: " + e);
      }
   }
}
```

o/p=>
Catch: java.lang.ArrayIndexOutOfBoundsException: 5.

2. Write a program which handles user defined exceptions.

```
class MyExcep extends Exception
{
   MyExcep()
   {
      super("Age must be 18 or above");
   }
}
```

```
class User_Excep
{
   public static void main(String arg[])
   {
      int age = 16;
      try
      {
         if(age < 18)
         {
            throw new MyExcep();
         }
         else
         {
            System.out.print("You are eligible to vote.");
         }
      }
      catch(MyExcep e)
      {
         System.out.print("Catch: " + e.getMessage());
      }
   }
}

o/p=>
Catch: Age must be 18 or above
```

3. Write a program to create thread by extending thread class and by implementing runnable interface.

```
//Thread creating by extending thread class
class Multi_Thred extends Thread
{
   public void run()
   {
      System.out.println("Thread " + Thread.currentThread().getId() + " is running");
   }
}

class Thr_Ext
{
   public static void main(String arg[])
   {
      Multi_Thred obj1 = new Multi_Thred();
      Multi_Thred obj2 = new Multi_Thred();
      Multi_Thred obj3 = new Multi_Thred();
      obj1.start();
      obj2.start();
      obj3.start();
   }
}
```

```
o/p=>
Thread 8 is running.
Thread 10 is running.
Thread 9 is running.


//Thread class creating by implementing runnable interface
class Multi_Thred implements Runnable
{
   public void run()
   {
      System.out.println("Thread " + Thread.currentThread().getId() + " is running.");
   }
}

class Thr_Imp
{
   public static void main(String arg[])
   {
      Multi_Thred obj1 = new Multi_Thred();
      Multi_Thred obj2 = new Multi_Thred();
      Multi_Thred obj3 = new Multi_Thred();

      Thread  a = new Thread(obj1);
      Thread  b = new Thread(obj2);
      Thread  c = new Thread(obj3);

      a.start();
      b.start();
      c.start();
   }
}

o/p=>
Thread 10 is running.
Thread 9 is running.
Thread 8 is running.
```

4. Write a program to implement thread priority.

```
class Multi_Thred extends Thread
{
   public void run()
   {
      System.out.println("Thread in running ID is: " + Thread.currentThread().getId() + ",
Name is: " + Thread.currentThread().getName() + ", Priority is: " +
Thread.currentThread().getPriority());
   }
}
```

```
class Thr_Priority
{
   public static void main(String arg[])
   {
      Multi_Thred obj1 = new Multi_Thred();
      Multi_Thred obj2 = new Multi_Thred();
      Multi_Thred obj3 = new Multi_Thred();

      obj1.setPriority(Thread.MIN_PRIORITY);
      obj2.setPriority(Thread.NORM_PRIORITY);
      obj3.setPriority(Thread.MAX_PRIORITY);

      obj1.start();
      obj2.start();
      obj3.start();
   }
}

o/p=>
Thread in running ID is: 10, Name is: Thread-2, Priority is: 10
Thread in running ID is: 9, Name is: Thread-1, Priority is: 5
Thread in running ID is: 8, Name is: Thread-0, Priority is: 1
```

5. Write a program of thread synchronization.

```
class Table
{
   synchronized void printTable(int n)
   {
      for(int i = 1; i <= 5; i++)
      {
         System.out.println(n * i);
      }
   }
}

class MyThread1 extends Thread
{
   Table t;
   MyThread1(Table t)
   {
      this.t = t;
   }
   public void run()
   {
      t.printTable(5);
   }
}
```

```
class MyThread2 extends Thread
{
   Table t;
   MyThread2(Table t)
   {
      this.t = t;
   }
   public void run()
   {
      t.printTable(100);
   }
}

class Thr_Synch
{
   public static void main(String arg[])
   {
      Table obj = new Table();
      MyThread1 obj1 = new MyThread1(obj);
      MyThread2 obj2 = new MyThread2(obj);
      obj1.start();
      obj2.start();
   }
}

o/p=>
5
10
15
20
25
100
200
300
400
500
```

6. Write a program which writes data into text file.

```
import java.io.*;

class Text_File
{

   public static void main(String arg[])
   {

      String s = "Hello World...!";
```

```
      try
      {
         FileWriter fw = new FileWriter("D:/Java/Text.txt");
         fw.write(s);
         fw.close();
         System.out.println("Successfully Data written in text file.");
      }
      catch(Exception e)
      {
         System.out.print("Catch: " + e);
      }
   }
}

o/p=>
Successfully Data written in text file.
```

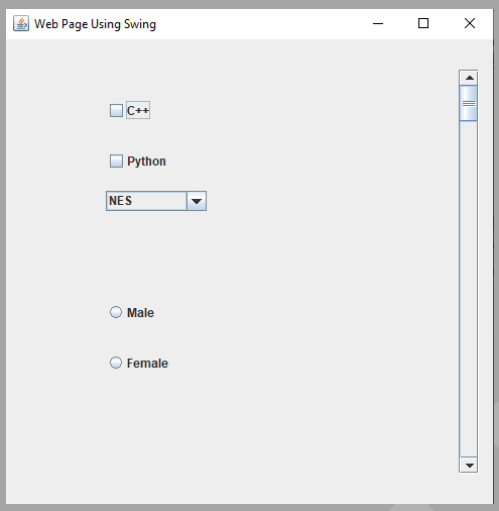
7. Write a program to copy one file into another file.

```
import java.io.*;

class Copy_File
{
   public static void main(String arg[])
   {
      String s = "Hello World...!";
      int ch;
      try
      {
         FileReader fr = new FileReader("D:/Java/Text.txt");
         while((ch = fr.read()) != -1)
         {
            System.out.print((char)ch);
         }
         fr.close();
      }
      catch(Exception e)
      {
         System.out.print("Catch: " + e);
      }
   }
}

o/p=>
Hello World...!
```

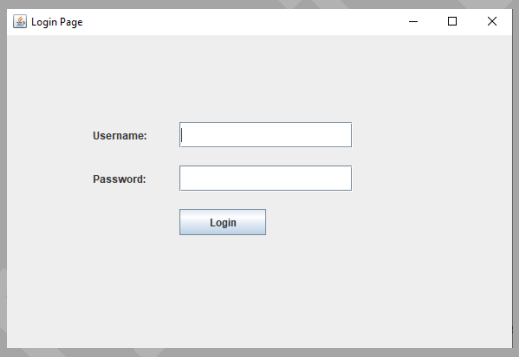
8. Write a swing program to create,
a) JScrollBar b)JCheckBox c)JComboBox d)JRadio

```
import javax.swing.*;

class Hybrid
{
   public static void main(String arg[])
   {
      JFrame frame = new JFrame("Web Page Using Swing");
      frame.setSize(500, 500);

      JScrollBar scroll = new JScrollBar();
      scroll.setBounds(450, 30, 20, 400);
      frame.add(scroll);

      // Adding Checkboxes
      JCheckBox box1 = new JCheckBox("C++");
      box1.setBounds(100, 50, 100, 40);
      frame.add(box1);

      JCheckBox box2 = new JCheckBox("Python");
      box2.setBounds(100, 100, 100, 40);
      frame.add(box2);

      // ComboBox for college selection
      String clg[] = {"NES", "SMS", "FBT", "SPT"};
      JComboBox box = new JComboBox(clg);
      box.setBounds(100, 150, 100, 20);
      frame.add(box);

      // Radio buttons for gender selection
      JRadioButton button1 = new JRadioButton("Male");
      button1.setBounds(100, 250, 100, 40);
      JRadioButton button2 = new JRadioButton("Female");
      button2.setBounds(100, 300, 100, 40);

      ButtonGroup group = new ButtonGroup();
      group.add(button1);
      group.add(button2);

      frame.add(button1);
      frame.add(button2);

      frame.setLayout(null);
      frame.setVisible(true);
   }
}
```

o/p=>



9. Write a swing program to create following GUI




```
import javax.swing.*;

class Login_Page extends JFrame
{
   Login_Page()
   {
      setTitle("Login Page");

      setSize(600, 400);

      // Username label
      JLabel userlabel = new JLabel("Username: ");
      userlabel.setBounds(100, 100, 100, 30);
      add(userlabel);

      // Username text field
      JTextField text = new JTextField();
      text.setBounds(200, 100, 200, 30);
      add(text);
```

```
        // Password label
        JLabel passlabel = new JLabel("Password: ");
        passlabel.setBounds(100, 150, 100, 30);
        add(passlabel);

        // Password field
        JPasswordField password = new JPasswordField();
        password.setBounds(200, 150, 200, 30);
        add(password);

        // Login button
        JButton button = new JButton("Login");
        button.setBounds(200, 200, 100, 30);
        add(button);

        setLayout(null);
        setVisible(true);
    }

    public static void main(String arg[])
    {
        new Login_Page();
    }
}
```

10. Write a java program which implement Arraylist class which use list interface.

```
import java.util.ArrayList;

class Arraylist_Inter
{
    public static void main(String arg[])
    {
        ArrayList<String> ar = new ArrayList<String>();

        ar.add("Java");
        ar.add("Python");
        ar.add("C++");
        ar.add(0, "JavaScript");

        System.out.println("\nInitial Elements in ArrayList: " + ar);

        ar.remove("Java");
        ar.set(2, "C#");

        System.out.println("\nCurrent Elements in ArrayList:");
        for (String a : ar)
```

```
      {
         System.out.println("- " + a);
      }

      System.out.println("\nList contains 'Python': " + ar.contains("Python"));

      System.out.println("Index of 'C#': " + ar.indexOf("C#"));

      System.out.println("List size: " + ar.size());

      ar.clear();
      System.out.println("\nList is empty: " + ar.isEmpty());
   }
}
```

o/p=>

```
Initial Elements in ArrayList: [JavaScript, Java, Python, C++]

Current Elements in ArrayList:
- JavaScript
- Python
- C#

List contains 'Python': true
Index of 'C#': 2
List size: 3

List is empty: true
```

11. Write a java program which implement Stack class which use list interface.

```
import java.util.Stack;

class Stack_Inter
{
   public static void main(String arg[])
   {
      Stack st = new Stack();

      st.push(10);
      st.push(20);
      st.push(30);

      System.out.println("\nInitial Elements in Stack: " + st);

      st.add(1, 15);
      st.set(2, 25);
```

```
        System.out.println("\nCurrent Elements in Stack:");
        for (Object a : st)
        {
           System.out.println("- " + a);
        }

        System.out.println("\nPopped Element: " + st.pop());
        System.out.println("\nStack contains 20: " + st.contains(20));
        System.out.println("Index of 25: " + st.indexOf(25));
        System.out.println("Stack size: " + st.size());

        st.clear();
        System.out.println("\nStack is empty: " + st.isEmpty());
    }
}

o/p=>
Initial Elements in Stack: [10, 20, 30]

Current Elements in Stack:
- 10
- 15
- 25
- 30

Popped Element: 30

Stack contains 20: false
Index of 25: 2
Stack size: 3

Stack is empty: true
```

12. Write a java program which implement HashSet class which use Set interface

```
import java.util.HashSet;
import java.util.Iterator;

class Hashset_Inter
{
    public static void main(String arg[])
    {

        HashSet<Integer> set = new HashSet<>();

        set.add(10);
        set.add(20);
```

```
      set.add(30);

      System.out.println("\nInitial Elements in HashSet: " + set);

      set.remove(10);

      System.out.println("\nCurrent Elements in HashSet:");
      Iterator<Integer> i = set.iterator();
      while (i.hasNext())
      {
         System.out.println("- " + i.next());
      }

      System.out.println("\nSet contains 20: " + set.contains(20));
      System.out.println("Set size: " + set.size());

      set.clear();
      System.out.println("Set is empty: " + set.isEmpty());
   }
}

o/p=>
Initial Elements in HashSet: [20, 10, 30]

Current Elements in HashSet:
- 20
- 30

Set contains 20: true
Set size: 2
Set is empty: true
```

13. Write a java program which implement LinkedHashSet class which use Set interface

```
import java.util.LinkedHashSet;

class LinkedHashSet_Inter
{
   public static void main(String arg[])
   {
      LinkedHashSet<Integer> set = new LinkedHashSet();

      set.add(10);
      set.add(20);
      set.add(30);
      System.out.println("\nOriginal LinkedHashSet: " + set);
```

```
        LinkedHashSet<Integer> cloneset = (LinkedHashSet<Integer>) set.clone();
        System.out.println("Cloned LinkedHashSet: " + cloneset);

        set.remove(10);

        System.out.println("\nCurrent Elements in Array:");
        Object arr[] = set.toArray();
        for (Object a : arr)
        {
           System.out.println("- " + a);
        }

        System.out.println("\nSet contains 20: " + set.contains(20));
        System.out.println("Set size: " + set.size());

        set.clear();
        System.out.println("Set is empty: " + set.isEmpty());
    }
}

o/p=>
Original LinkedHashSet: [10, 20, 30]
Cloned LinkedHashSet: [10, 20, 30]

Current Elements in Array:
- 20
- 30

Set contains 20: true
Set size: 2
Set is empty: true
```

14. Write a java program which implement HashMap class which use Map interface

```
import java.util.HashMap;
import java.util.Map;

class HashMap_Inter
{
    public static void main(String arg[])
    {
        HashMap<Integer, String> map = new HashMap<>();

        map.put(1, "Java");
        map.put(2, "Python");
        map.put(3, "C++");
```

```
        System.out.println("\nInitial Elements in HashMap: " + map);

        map.put(2, "JavaScript");
        map.remove(1);

        System.out.println("\nCurrent Elements in HashMap:");
        for (Map.Entry a : map.entrySet())
        {
           System.out.println("- Key: " + a.getKey() + ", Value: " + a.getValue());
        }

        System.out.println("\nValue for key 3: " + map.get(3));
        System.out.println("\nContains key 3: " + map.containsKey(3));
        System.out.println("Contains value 'C++': " + map.containsValue("C++"));
        System.out.println("\nAll Keys: " + map.keySet());
        System.out.println("All Values: " + map.values());
        System.out.println("All Key-Value Pairs: " + map.entrySet());
        System.out.println("\nSize of HashMap: " + map.size());

        map.clear();
        System.out.println("\nMap is empty: " + map.isEmpty());
    }
}

o/p=>
Initial Elements in HashMap: {1=Java, 2=Python, 3=C++}

Current Elements in HashMap:
- Key: 2, Value: JavaScript
- Key: 3, Value: C++

Value for key 3: C++

Contains key 3: true
Contains value 'C++': true

All Keys: [2, 3]
All Values: [JavaScript, C++]
All Key-Value Pairs: [2=JavaScript, 3=C++]

Size of HashMap: 2

Map is empty: true
```

15. Write a java program which implement TreeMap class which use Map interface

```
import java.util.TreeMap;

class TreeMap_Inter
{
   public static void main(String arg[])
   {
      TreeMap<Integer, String> map = new TreeMap<>();

      map.put(3, "C++");
      map.put(1, "Java");
      map.put(2, "Python");

      System.out.println("TreeMap: " + map);
      System.out.println("First key: " + map.firstKey());
      System.out.println("Last key: " + map.lastKey());
      System.out.println("Higher than 2: " + map.higherKey(2));
      System.out.println("Keys: " + map.keySet());
      System.out.println("Values: " + map.values());
      System.out.println("Entry Set: " + map.entrySet());    }
}

o/p=>
TreeMap: {1=Java, 2=Python, 3=C++}
First key: 1
Last key: 3
Higher than 2: 3
Keys: [1, 2, 3]
Values: [Java, Python, C++]
Entry Set: [1=Java, 2=Python, 3=C++]
```
