Name:

Enrolment No:

UPES

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

End Semester Examination, May 2021

Programme Name: B.Tech –MECHATRONICS		**Semester**	**: VIII**
Course Name	**: JAVA Programming**	**Time**	**: 3 Hr**
Course Code	**: CSEG 4002P**	**Max. Marks**	**: 100**

Instructions:

1. **Attempt all the questions (Theory, Numerical, Case study etc.)**
2. **Attempt all questions serially as per Question paper.**
3. **Answer should be neat and clean. Draw a free hand sketch for circuits/tables/schematics wherever required.**
4. **Scan the required answer script and check the resolution carefully before uploading. No other mode of submission is acceptable.**
5. **You are expected to be honest about each attempt which you make to progress in life**

SECTION A [5x6]

S. No.		Marks	CO
Q1			**CO1**
I	**The JRE performs the following main tasks:**	**2**	
(A)	○ Executes code		
(B)	○ Verifies code		
(C)	○ Provides runtime environment		
(D)	○ None of the above		
II	Method overriding is the example of compile-time polymorphism **(True/False).**	**1**	

<table>
<tr>
<td><b>III</b></td>
<td>

Mention the output:

```
class Evaluation
{
 void act(int  n)
{
  int act=1;
  for(int i=1;i<=n;i++)
.{
   act=act*i;
  }
 System.out.println("Calculated output is "+act);
}
public static void main(String args[])
   {
      new Evaluation().act(5);//calling method with anonymous object
    }

}
```

</td>
<td><b>2</b></td>
<td></td>
</tr>
<tr>
<td><b>Q2</b></td>
<td>

Write the output of the following code:

```
class Student
{
 int rollno;
 String name;
 void insertRecord(int r, String n)
{
  rollno=r;
  name=n;
 }
      void displayInformation()
          {
              System.out.println(rollno+" "+name);
           }
  }


}
class TestStudent4
     {
 public static void main(String args[])
{
  Student s1=new Student();
```

</td>
<td><b>5</b></td>
<td><b>CO2</b></td>
</tr>
</table>

```
  Student s2=new Student();
  s1.insertRecord(374685812,"Arjun");
  s2.insertRecord(520897741,"Vipul");
  s1.displayInformation();
  s2.displayInformation();
  }
}
```

Q3

I

What should be the execution order, if a class has a method, static block, instance block, and constructor, as shown below?

```
public class First_C {
      public void myMethod()
    {
    System.out.println("Method");
    }
    {
    System.out.println(" Instance Block");
    }
    public void First_C()
    {
        System.out.println("Constructor ");
     }
       static {
      System.out.println("static block");
    }
    public static void main(String[] args) {
    First_C c = new First_C();
    c.First_C();
    c.myMethod();
  }
}
```

(A) Instance block, method, static block, and constructor

(B) Method, constructor, instance block, and static block

(C) Static block, method, instance block, and constructor

(D) Static block, instance block, constructor, and method

3+2

CO1

II **An interface with no fields or methods is known as a ______.**

(A) Runnable Interface

(B) Marker Interface

(C) Abstract Interface

(D) Char-Sequence Interface

Q4

I **Guess the output:**

```
public class Test {
public static void main(String[] args) {
   int count = 1;
   while (count <= 15) {
   System.out.println(count % 2 == 1 ? "***" : "+++++");
   ++count;
      }     // end while
   }      // end main
 }
```

(A) 15 times ***

(B) 15 times +++++

(C) 8 times *** and 7 times +++++

(D) Both will print only once

3+2

CO1

II **In which memory a String is stored, when we create a string using new operator?**

(A) Stack

(B) String memory

(C)	Heap memory		
(D)	Random storage space		
Q5		2+2+1	CO1
I	**If a thread goes to sleep?**		
(A)	It releases all the locks it has.		
(B)	It does not release any locks.		
(C)	It releases half of its locks.		
(D)	It releases all of its lock except one.		

II	`abstract class MyFirstClass`
	`{`
	`    abstract num (int a, int b) {  }`
	`}`		

(A)	No error		
(B)	Method is not defined properly		
(C)	Constructor is not defined properly		
(D)	Extra parentheses		
III	**How many threads can be executed at a time?**		
(A)	Only one thread		
(B)	Multiple threads		
(C)	Only main (main() method) thread		
(D)	Two threads		
Q6		3+2	CO3
I	**Which of the following is true about Servlets?**		

(A) Servlets execute within the address space of web server;

(B) Servlets are platform-independent because they are written in java;

(C) Servlets can use the full functionality of the Java class libraries;

(D) Servlets execute within the address space of web server, platform independent and uses the functionality of java class libraries;

II Go through a Java program to illustrate the fact that runtime polymorphism cannot be achieved by data members. **Expected output is:**

```
class A

{

        int x = 10;

}

class B extends A

{

        int x = 20;

}

// Driver class

public class Test

{

        public static void main(String args[])

        {

                A a = new B();

                System.out.println(a.x);

        }

}
```

(A) 20

(B) 10

(C) Compilation Error

(D) None of the above

SECTION B [10 x 5]			
Q7	**Answer the following :** **a)** Why do we use interface? **b)** Advantages of a Java Servlet. **c)** Why Java doesn’t support Multiple Inheritance? Justify your answer with the adequate code.	**4+3+3**	**CO2**
Q8 **(a)**	**Mention the output of the following code:**	**5+5**	**CO3**

```
class A
{
        public void print_A() { System.out.println("Class A"); }
}

class B extends A
{
        public void print_B() { System.out.println("Class B"); }
}

class C extends A
{
        public void print_C() { System.out.println("Class C"); }
}

class D extends A
{
        public void print_D() { System.out.println("Class D"); }
}

// Driver Class
public class Test
{
        public static void main(String[] args)
        {
                B obj_B = new B();
                obj_B.print_A();
                obj_B.print_B();

                C obj_C = new C();
                obj_C.print_A();
                obj_C.print_C();

                D obj_D = new D();
                obj_D.print_A();
                obj_D.print_D();
        }
}
```

(b)

Elucidate the following built-in Exceptions in Java:

a) NullPointerException
b) NumberFormatException
c) StringIndexOutOfBoundsException
d) FileNotFoundException
e) IOException

Q9

(a)

Guess the output of the following code:

[4+6] **CO3**

```
abstract class A {

    abstract void m1();

    void m2()

    {

        System.out.println("This is "

                           + "a concrete method.");

    }

}



// concrete class B

class B extends A {

    void m1()

    {

        System.out.println("B's "

                           + "implementation of m1.");
    }
}

// Driver class
public class AbstractDemo {
    public static void main(String args[])
    {
        B b = new B();
        b.m1();
        b.m2();
    }
}
```

(b)

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

// A class to represent a student.
class Student {
        int rollno;
        String name, address;

        // Constructor
        public Student(int rollno, String name, String address)
        {
                this.rollno = rollno;
                this.name = name;
                this.address = address;
        }

        // Used to print student details in main()
        public String toString()
        {
                return this.rollno + " " + this.name + " "
                    + this.address;
        }
}

class Sortbyroll implements Comparator<Student> {
        // Used for sorting in ascending order of roll number
        public int compare(Student a, Student b)
        {
                return a.rollno - b.rollno;
        }
}

class Sortbyname implements Comparator<Student> {
        // Used for sorting in ascending order of
        // name
        public int compare(Student a, Student b)
        {
                return a.name.compareTo(b.name);
        }
}
```

```
// Driver class

class Main {

        public static void main(String[] args)

        {

                ArrayList<Student> ar = new ArrayList<Student>();

                ar.add(new Student(111, "bbbb", "london"));

                ar.add(new Student(131, "aaaa", "nyc"));

                ar.add(new Student(121, "cccc", "jaipur"));


                System.out.println("Unsorted");

                for (int i = 0; i < ar.size(); i++)

                        System.out.println(ar.get(i));


                Collections.sort(ar, new Sortbyroll());


                System.out.println("\nSorted by rollno");

                for (int i = 0; i < ar.size(); i++)

                        System.out.println(ar.get(i));


                Collections.sort(ar, new Sortbyname());

                System.out.println("\nSorted by name");

                for (int i = 0; i < ar.size(); i++)

                        System.out.println(ar.get(i));

        }

}
```

Endeavor to suggest the possible outcome of the aforementioned code;

5+5

Q10

(a)

```
import java.io.*;

// ……………………..C1
```

5+5

```
interface In1
{
// ……………………..C2
final int a = 10;

// ……………………..C3
void display();
}

// ……………………..C4
class TestClass implements In1
{
// ……………………..C5
public void display()

{

System.out.println("Geek");

}
public static void main (String[] args)
{
    TestClass t = new TestClass();
     t.display();
     System.out.println(a);
 }
```

Write the output of the following code and Fill in the Boxe(s) indicated with ……………………..C

CO2

(b)	**Write a Java code for thread creation by implementing the Runnable Interface.**		
Q11	**Brief about:** i. ‘final’ keyword ii. ‘extends’ keyword iii. Non-static method(s) iv. ‘this’ keyword v. JDK vi. Abstract class vii. Class PATH viii. Run time Polymorphism ix. Generic Servlets x. Exception Handling	**10**	**CO3**
SECTION ‘C’[20 MARKS]			
Q12	Write a program in JAVA language to implement the following instruction(s): a) Invoke immediate parent class constructor using “**super()**”; b) Return the current class instance using “**this**” keyword; c) When implementing an interface we force the class to implement its methods and if it is not implementing it then we need to declare that class abstract. d) Resolve Ambiguity e) Demonstrate the code reusability using “**extends**” keyword;	**4+4+4 +4+4**	**CO2**