Name:

Enrolment No:

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

End Semester Examination, December 2020

Programme Name: B.Tech –ECE
Course Name : Programming in JAVA SDR
Course Code : ECEG-4018P

Semester : VII
Time : 3 Hr
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 JVM performs the following main tasks:**	**2**	
(A)	○ Loads code		
(B)	○ Verifies code		
(C)	○ Executes code		
(D)	○ Provides runtime environment		
(E)	○ All of the above		
II	Method overriding is the example of run time polymorphism **(True/False).**	**1**	

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

Mention the output:

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

}
```

</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Q2</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();
  Student s2=new Student();
```

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

	s1.insertRecord(111,"Karan");
	s2.insertRecord(222,"Aryan");
	s1.displayInformation();
	s2.displayInformation();
	}
	}		

Q3 **I** **(A)** **(B)** **(C)** **(D)** **II** **(A)** **(B)** **(C)** **(D)** **III** **(A)** **(B)** **(C)** **(D)**	Which of the following for loop declaration is not valid? for (int i = 99; i >= 0; i / 9) for (int i = 7; i <= 77; i += 7) for (int i = 20; i >= 2; - -i) for (int i = 2; i <= 20; i = 2* i) Which package contains the Random class? java.util package java.lang package java.awt package java.io package An interface with no fields or methods is known as a ______. Runnable Interface Marker Interface Abstract Interface Char-Sequence Interface	**2+2+1**	**CO1**
Q4 **I** **(A)** **(B)**	Which option is false about the final keyword? A final method cannot be overridden in its subclasses. A final class cannot be extended.	**3+2**	**CO1**

(C)	A final class cannot extend other classes.		
(D)	A final method can be inherited.		
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 **I**	Which keyword is used for accessing the features of a package?	**2+2+1**	**CO1**
(A)	package		
(B)	import		
(C)	extends		
(D)	export		

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) **(D)**	Only main (main() method) thread Two threads		
Q6 **I** **(A)** **(B)** **(C)** **(D)** **II** **(A)** **(B)** **(C)** **(D)**	Which of the following is true about Servlets? Servlets execute within the address space of web server; Servlets are platform-independent because they are written in java; Servlets can use the full functionality of the Java class libraries; Servlets execute within the address space of web server, platform independent and uses the functionality of java class libraries; Which of the following option leads to the portability and security of Java? Bytecode is executed by JVM The applet makes the Java code secure and portable Use of exception handling Dynamic binding between objects	**3+2**	**CO3**
SECTION B [10 x 5]			
Q7	**Answer the following :** **a)** Is main method compulsory in JAVA? **b)** How are parameters passed in JAVA? **c)** Why Java doesn't support Multiple Inheritance? Justify your answer with the adequate code. **d)** Advantages of a Java Servlet	**2+2+3 +3**	**CO2**
Q8	**Elucidate the following built-in Exceptions in Java:** a) NoSuchFieldException b) NullPointerException c) NumberFormatException	**10**	

d) RuntimeException
e) StringIndexOutOfBoundsException
f) FileNotFoundException
g) IOException
h) InterruptedException
i) ArrayIndexOutOfBoundsException
j) ClassNotFoundException

CO3

Q9

Guess the output of the following code:

```
// Java code for thread creation by extending the Thread class

class MultithreadingDemo extends Thread
{
    public void run()
    {
        try
        {
            // Displaying the thread that is running
            System.out.println ("Thread " +
                  Thread.currentThread().getId() +
                  " is running");


        }
        catch (Exception e)
        {
            // Throwing an exception
            System.out.println ("Exception is caught");
        }
    }
}

// Main Class
public class Multithread
{
    public static void main(String[] args)
    {
        int n = 8; // Number of threads
        for (int i=0; i<n; i++)
        {
            MultithreadingDemo object = new MultithreadingDemo();
            object.start();
        }
    }
}
```

****Rewrite the above code for Thread creation by implementing the Runnable Interface**

[4+6]

CO3

Q10

Let’s consider the real world example of vehicles like bicycle, car, bike………, they have common functionalities. So we make an interface and put all these common functionalities. And lets Bicycle, Bike, car ….etc implement all these functionalities in their own class in their own way:

10

```
import java.io.*;

interface Vehicle
{

    // ……………………………..C1
    void changeGear(int a);
    void speedUp(int a);
    void applyBrakes(int a);
}

class Bicycle implements Vehicle{

    int speed;
    int gear;

     // ………………………C2
    @Override
    public void changeGear(int newGear){

        gear = newGear;
    }

        @Override
    public void speedUp(int increment){

        speed = speed + increment;
    }

    // ………………………………C3
    @Override
    public void applyBrakes(int decrement){

        speed = speed - decrement;
    }

    public void printStates() {
         System.out.println("speed: " + speed
              + " gear: " + gear);
    }
}

class Bike implements Vehicle {

    int speed;
    int gear;

    // ……………………..C4
    @Override
    public void changeGear(int newGear){

        gear = newGear;
    }

    // ……………………..C5
```

CO2

```
    @Override
    public void speedUp(int increment){

        speed = speed + increment;
    }

    // ……………………..C6
    @Override
    public void applyBrakes(int decrement){

        speed = speed - decrement;
    }

    public void printStates() {
         System.out.println("speed: " + speed
             + " gear: " + gear);
    }

}
class GFG {

    public static void main (String[] args) {

        // ……………………..C7

        Bicycle bicycle = new Bicycle();
        bicycle.changeGear(2);
        bicycle.speedUp(3);
        bicycle.applyBrakes(1);

        System.out.println("Bicycle present state :");
        bicycle.printStates();

        // ……………………..C8
        Bike bike = new Bike();
        bike.changeGear(1);
        bike.speedUp(4);
        bike.applyBrakes(3);

        System.out.println("Bike present state :");
        bike.printStates();
    }
}
```

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

Q11	**Brief about:** i. ‘super’ keyword ii. ‘extends’ keyword iii. static method(s) iv. ‘this’ keyword v. JVM	**10**	**CO3**

	vi. Abstract class vii. Class PATH viii. Run time Polymorphism ix. Generic Servlets x. Exception Handling		
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) Implement the constructor overloading; d) Final methods with the help of Abstract class; e) Demonstrate the code reusability using “**extends**” keyword;	**4+4+4** **+4+4**	**CO2**