Title Student

Q1.

a) Create a java class called Student with the foll. details as variables within it.

1) USN
2) Name
3) Branch
4) Phone

Write a Java program to create 'n' student objects & print the USN, Name, Branch, & Phone of the objects with suitable headings.

o/p

```
import .java.util.Scanner;
public class Student
{
    String Name, USN, Branch, Number;
    int i, N;
```

Signature of Teacher incharge

```
ublic Student()
{
  this.Name = Name;
  this.USN = USN;
  this.Branch = Branch;
  this.Number = Number;
}
Void read()
{
  System.out.println("Enter no. of students : ");
  Scanner s5 = new Scanner(System.in);
  N = s5.nextInt();

  for(i = 0; i<=N; i++)
  {
    System.out.println("Enter details of student" +i+ ":");
    System.out.println("Enter Name : ");
    Scanner S1 = new Scanner(System.in);
    Name = S1.nextLine();

    System.out.println("Enter USN : ");
    Scanner S2 = new Scanner(System.in);
    USN = s2.nextline();
```

Dept./Lab DOPS Class B-1 Expt./No. 1(a)

Title Student

```
System.out.println("Enter Branch: ");
Scanner s3 = new Scanner(System.in);
Branch = s3.nextLine();

System.out.println("Enter Number: ");
Scanner s4 = new Scanner(System.in);
Number = s4.nextLine();

}

}

void display()
{
  for(i=0; i<N; i++)
  System.out.println("\t" + Name + "\t" + USN + "\t"
                     + Branch + "\t" + Number);
}
```

Signature of
Teacher incharge

```
public static void main(String[] args)
{
    student s1;
    s1 = new Student();
    s1.read();
    System.out.println("Student details are: ");
    System.out.println("\t Name: \t USN: \t Branch: \t
                                        Number: \t");

    s1.display();
    }
}
```

Date 2/2/18 Name

Dept./Lab OOPS Class B-1 Expt./No. 1(b)

Title Rectangle

Q1.

b) Write a program in Java with class Rectangle with the data fields width, length, area and color. The length, width and area are of double type & color is of string type. The methods are set_length(), set_width(), set_color() & find area(). Create two object of Rectangle and compare their area & color. If area & color both are the same for the object then display "Matching Rectangle", otherwise display "Non-Matching Rectangle".

```
import java.util.Scanner;

public class Rectangle
{
  double length, width, area;
  String color;
```

Signature of Teacher incharge

```
Scanner s1 = new Scanner(System.in);

void set_length ()
{
  System.out.println("Enter length: ");
  length = s1.nextDouble();
}
void set_width()
{
  System.out.println("Enter width: ");
  width = s1.nextDouble();
}
void set_color()
{
  System.out.println("Enter color: ");
  color = s1.next();

void find_area();
{
  area = length * breadth;
}
```

OBSERVATION / DATA SHEET

Date 2/2/18 Name Gautam Patil

Dept./Lab OOPS Class B-1 Expt./No. 1(6)

Title Rectangle

```
public static void main( String[] args)
{
  Rectangle r1 = new Rectangle();
  System.out.println("Enter details for Rectangle 1: ");
  r1.set_length();
  r1.set_width();
  r1.set_color();
  r1.find_area();
  Rectangle r2 = new Rectangle();
  System.out.println("Enter details for Rectangle 2: ");
  r2.set_length();
  r2.set_width();
  r2.set_color();
  r2.find_area();
```

Signature of
Teacher incharge

```
if((r1.color.equals(r2.color) && (r1.area ==
                                        r2.area
    System.out.println("Matching Rectangles.");
else
    System.out.println("Non-Matching Rectangles");
```

Title Overload constructor

Q1.

a) Write a java program to overload constructor & method.

```
public class constructor
{
  int i;
  constructor()
  {
    i=0;
    System.out.println("\n Non-Parameterised: " + i);
  }
  constructor(int h)
  {
    i=h;
    System.out.println("\n Parameterised: " + i);
  }
  void overload()
  {
```

Signature of Teacher incharge

ram to overload constru...

```
System.out.println("\n No parameter fn.");
}
void overload(int h)
{
  System.out.println("\n You entered : " +h);
}
public static void main(String[] args)
{
  constructor a = new constructor();
  Constructor b = new constructor();
  a.overload();
  b.overload(h: 6);
}
}
```

02.

a) Write a program in Java to create a Player class. Inherit the classes Cricket_Player, Football_Player & Hockey_Player from Player class.

```
public class Player
{
  String Name;
  int Age;

  Player (String a, int b)
  {
    Name = a;
    Age = b;
  }

  Void Show ( )
  {
    System.out.println("\n Name: " + Name + "\n Age" + Age);
  }
```

o/p screenshot

Signature of
Teacher incharge

```
class Cricket_Player extends Player
{
    String type;
    public Cricket_Player(String a, int b, String c)
    {
        super(a,b);
        type = c;
    }
    void show()
    {
        super.show();
        System.out.println("Type: " +type);
    }
}

class Football_Player extends Player
{
    String type;
    public Football_Player (String a, int b, String
    {
        super(a,b);
        type = c;
    }
```

```
    void Show()
    {
      super.Show();
      System.out.println("Type: "+type);
    }
}

class Hockey_Player extends Player
{
    String type;
    public Hockey_Player(String a, int b, String c)
    {
      ~~super.Show();~~
      super(a,b);
      type = c;
    }
```

Signature of Teacher incharge

Step 12: Defin
a. Call

Step 13: Defin

Step 14: Creat

Step 15: Displ

Step 16: End.

```
    void show()
    {
        super.show();
        System.out.println("Type:" + type);
    }
}

class Player_Main
{
    public static void main(String [] args)
    {
        Cricket_Player CP = new Cricket_Player
                            ("Hero", 19, "Cricket
        Football_Player FP = new Football_Player
                            ("Abhi", 20, "Football
        Hockey_Player HP = new Hockey_Player
                            ("Mani", 21, "Hockey"
        CP.show();
        FP.show();
        HP.show();
    }
```

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 15/2/18 Name Gautam Pathak

Dept./Lab OOPS Class B-1 Expt./No. 2(b)

Title Trunk calls

Q2.

b) Consider the trunk calls of a telephone exchange. A trunk call can be ordinary, urgent or lightning. The charges depend on the duration and the type of the call. Write a program using the concept of polymorphism in Java to calculate the charge.

```
import .Java .util. Scanner;

public class Call
{
    String type;
    float dur;

    float methodrate ()
    {
        if (type. compareTo ("urgent") == 0)
            return 4.5f;
```

Signature of Teacher incharge

```
else if (type.compareTo("lightning") == 0)
    return 3.5f;

else
    return 5f;

}

}

class bill extends Call
{
    double amount;
    String DataInputStream;
    bill()
    {
        String DataInputStream = null;
    }
    Scanner s1 = new Scanner(System.in);
    void read()
    {
        System.out.println("Enter duration:");
        dur = s1.nextFloat();
        System.out.println("Enter type:");
        type = s1.next();
    }
```

OUTPUT:

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

ate 15/2/18 Name Gautam Pathak

ept./Lab OOPS Class B-1 Expt./No. 2 (b)

tle Trunk calls

```
void calculate()
{
  if (dur < 1.5)
    amount = methodrate() * dur + 3.5f;
  else if (dur < 3)
    amount = methodrate() * dur + 2.5f;
  else if (dur < 5)
    amount = methodrate() * dur + 4.5f;
  else
    amount = methodrate() * dur + 5f;
}

void print()
{
  System.out.println("\n Type : "+type+ "\n Duration: "
                     +dur+ "\n Amount: "
                     +amount);
}
```

Signature of
Teacher incharge

```
class Trunk_Calls
{
  public static void main (String[ ] args)
  {
    bill B = new bills( );
    B.read( );
    B.calculate( );
    B.print( );
  }
}
```

Outputt →

Enter duration:
50
Enter type:
urgent

Type: urgent
Duration: 50.0
Amount: 2300

Enter duration: 4
Enter type: lightning

Type: lightning
Duration: 4.0
Amount: 18.5

Q2.

a) Design a superclass called Staff with details as Staff Id, Name, Phone, Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a Java program to read & display at least 3 staff objects of all three categories.

```
public class staff
{
    int StaffId;
    long Phone, Salary;
    String Name;

    public void set_StaffId( int StaffId)
    {
        this.StaffId = StaffId;
    }
```

Signature of
Teacher incharge

```
public void set_Phone (long Phone)
{
    this.Phone = Phone;
}
public void set_Salary (long Salary)
{
    this.Salary = Salary;
}
public void set_Name (String Name)
{
    this.Name = Name;
}

@Override
public String toString ()
{
    return "Staff" +
            "StaffId =" + StaffId +
            "Phone =" + Phone +
            "Salary =" + Salary +
            "Name =" + Name +;
}
}
```

```
class Teaching extends Staff
{
    String Domain, Publications;

    @Override
    public String toString()
    {
        return "Teaching :" +
               "\n StaffId =" + StaffId +
               "\n Place =" + Place +
               "\n Salary=" + Salary +
               "\n Name =" + Name +
               "\n Domain =" + Domain +
               "\n Publications =" + Publications;
    }
```

Signature of Teacher incharge

```
public void set_Domain( String Domain)
{
  this.Domain = Domain;
}
public void set_Publications ( String Publications)
{
  this.Publication = Publications;
}
}

class Technical extends Staff
{
  String skills;

  @Override
  public String toString ()
  {
    return "\n Technical : " +
           "\n StaffId = " + StaffId +
           "\n Place = " + Place +
           "\n Salary = " + Salary +
           "\n Name = " + Name +
           "\n Skills = " + skills;
  }
```

```
public void set_Skills (String skills)
{
    this.skills = skills;
}
}
class Contract extends Staff
{
    String Period;
    @Override
    public string toString()
    {
        return "\n Contract" +
                "\n Period = " + Period + ";
    }

    public void set_Period (String Period)
    {
        this.Period = Period;
    }
}
```

Signature of Teacher incharge

```
class Staff - Main
{
public static void main (String[ ] args)
{
Teaching T1 = new Teaching ();
T1. set_Name ("Dhiru");
T1. set_Phone (9876543);
T1. set_Salary (55000);
T1. set_Staff Id ("Work Group");
T1. set_Publications ("Bharat");
System. out. println (T1. to String ());

Technical T2 = new Technical ();
T2. set_Name ("Rahul");
T2. set_Phone (6059785);
T2. set_Salary (65000);
T2. set_Staff Id (6785);
T2. set_skills ("Parkour");
System. out. println (T2. to String ());

Contract C = new Contract ();
C. set_Name ("Lorman");
C. set_Phone (890870997);
```

```
C.set-Salary (75000);
C.set-StaffId (8970);
C.set-Period ("10 Hours");
System.out.println(C.toString());
}
}
```

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 9/8/18 Name Gautam Pathak

Dept./Lab OOPS Class 4B-1 Expt./No. 3(a)

Title Package & Interface

3(a).

Write a program to make a package Balance in which has Account class with Display-Balance method in it. Import balance package in another program to access Display_Balance method of Account class.

Main.java →

```
package.com.Company

public class Main
{
  public static void main(String[] args)
  {
    Balance.Account a1 = new Balance.Account();
    a1.read();
    a1.Display_balance();
  }
}
```

Signature of Teacher incharge

Account.java →

```
import java.util.Scanner;
public class Account
{
  int acc_no;
  String name;
  float balance;
  public void read()
  {
    Scanner s1 = new Scanner(System.in);
    System.out.println("Enter details of account:");
    System.out.println("Name:");
    name = s1.nextLine();
    System.out.println("Account No:");
    acc_no = s1.nextInt();
    System.out.println("Balance:");
    balance = s1.nextFloat();
  }
  public void Display_balance()
  {
    System.out.println("Details of the account:");
    System.out.println("\n Account No.:" + acc_no + "\n Name:
                       + name + "\n Balance:" + balance
  }
}
```

3 b) Create the dynamic stack by implementing the interface that define push() and pop() methods.

Dynstack.java

```
public class Dynstack implements Stacks top
{
  int tos;
  int size;
  int [] stk;
  Dynstack(int size)
  {
    tos = -1;
    this.size = size;
    stk = new int [size];
}
public void push (int item)
```

Signature of
Teacher incharge

```
{
  if (tos >= size)
    System.out.println("overflow");
  else
  {
    this.tos++;
    stk[tos] = item;
  }
}
public int pop()
{
  if(tos<0)
    return -1;
  else
  {
    this.tos--;
    return stk[tos+1];
  }
}
}
```

Main.java

```
import java.util.Scanner;
```

Dept./Lab OOPS Class 4B-1 Expt./No. 3(b)

Title

```
interface Stacktop
{
  int pop();
  void push(int item);
}

public class Main
{
  public static void main(String[] args)
  {
    int size, a;
    Scanner s1 = new Scanner(System.in);
    System.out.println("Enter size of stack:");
    siz = s1.nextInt();
    System.out.println("Size :" +siz);
    Stacktop mystk = new Dynstack(siz);
    System.out.println("Enter the
                        element :");
```

Signature of Teacher incharge

```
for (int i=0; i<siz; i++)
 {
   a = s1.nextInt();
   mystk.push(a);
 }
System.out.println("Contents of Stack (displayed by popping)");
for(int i=0; i<siz; i++)
 {
   a = mystk.pop();
   if (a == -1) System.out.println("Underflow");
   else System.out.println(a);
 }
Dynstack d1 = new Dynstack(siz);
mystk = d1;
System.out.println("Enter elements");
for(int i=0; i<siz; i++)
 {
   a = s1.nextInt();
   mystk.push(a);
 }
System.out.println("Contents of Stack (displayed by pop
for(int i=0; i<siz; i++)
 {
```

```
a = mystk.pop();
if (a == -1) System.out.println("Underflow");
else System.out.println(a);
}
}
```

Q4. On a single track two vehicles are running. If vehicles are going in same direction then is no problem. If the vehicles are running in different direction then is a chance of collision. To avoid collision write a Java program using exception handling. You are free to take necessary assumptions.

Collision.java

```
public class Collision extends Exception
{
  public Collision (String s)
   {
     super(s);
   }
}
```

PTS

Signature of
Teacher incharge

Two Vehicles.java

```
import com.company.Collision;
import java.util.Scanner;

public class TwoVehicles
{
    public static void main(String[] args)
    {
        String t1 = null;
        String t2 = null;
        Scanner s1 = new Scanner(System.in);
        try {
        System.out.println("Enter the direction of Vehicle 1 :");


        t1 = nextLine();
        System.out.println("Enter the direction of Vehicle 2 :");
        t2 = nextLine();
        if(!t1.equals(t2))
            throw new Collision("Vehicle 2 has to go in " + t2 +
                                " direction");

    }
```

```
catch (Collision e)
{
  System.out.println(e);
  t1 = t2;
  System.out.println("\n Collision has been avoided ");
}
System.out.println(" Direction of vehicle 1 : " + t1);
System.out.println(" Direction of vehicle 2 : " + t2);
}
```

O/P

Enter the direction of vehicle 1 :
left
Enter the direction of vehicle = 2 :
~~right~~ left

Signature of

Date 22/3/18 Name [illegible]

Dept./Lab OOPS Class 4B-1 Expt./No. 5(a)

Title Multithreading

5a) Write a java program to create five thread with diff. priorities. Send two thread of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting.

```
public class Thread_Main extends Thread
{
  public static final int threadNumber = 5;
  public static void main(String[] args) throws
    InterruptedException {
  Thread[] T1 = new Thread[threadNumber];
  for(int i=0; i< threadNumber; i++)
    {
      T1[i] = new Thread();
      T1.setPriority[i+1];
    }
```

Signature of Teacher incharge

```
T1[0].sleep(1000);
if(T1[0].isAlive())
  System.out.println("Thread 1 is alive.");
else
  System.out.println("Thread 1 is not alive.");
T1[1].start();
if(T1[i].isAlive())
  System.out.println("Thread 2 is alive");
else
  System.out.println("Thread 2 is not alive");
T1[2].sleep(1000);
if(T1[2].isAlive())
  System.out.println("Thread 3 is alive");
else
  System.out.println("Thread 3 is not alive");
T1[3].start();
if(T1[3].isAlive())
  System.out.println("Thread 4 is alive");
else
  System.out.println("Thread 4 is not alive");
```

```
T1[4].sleep(1000);
if (T1[4].isAlive())
 System.out.println("Thread 5 is alive");
else
 System.out.println("Thread 5 is not alive");
}
}
```

5.b) Write a multithreaded Java program to implement producer consumer problem.

Producer Consumer.java

```
public class ProducerConsumer
{
public static void main(String[] args) throws
InterruptedException
 {
  PC p = new PC();
 Thread T1 = new Thread(new Runnable()
  {
  public void run()
    {
     try { p.produce(); }
     catch(InterruptedException e)
      {
        System.out.println(e);
      }
```

Signature of Teacher incharge

```
Thread T2 = new Thread( new Runnable()
  {
    public void run()
    {
      try { t.consume(); }
      catch(InterruptedException e)
      {
        System.out.println(e);
      }
    }
  });
    T1.start();
    T2.start();
    T1.join();
    T2.join();
  }
}
```

~~class P~~

PC.java

```
class PC
{
  LinkedList<Integer> list = new LinkedList<>();
  int capacity = 2;
```

```
public void produce() throws InterruptedException
{
int value=0;
while(true)
{
 synchronized(this)
  {
   while(list.size==capacity)
    wait();
   System.out.println("Producer produced"+value);
   list.void(value++); notify();
   Thread.sleep(500);
  }
}
}
public void consume() throws InterruptedException
{
 while(true)
  {
   synchronized(this)
```

Signature of
Teacher incharge

```
{
while (list.size() == 0) wait();
int ele = list.removeFirst();
System.out.println("Consumer consumed " + ele);
notify();
Thread.sleep(500);
}
}
}
```

o/p

Producer produced 0

R.V. COLLEGE OF ENGINEERING

OBSERVATION / DATA SHEET

Date 22/3/18 Name Gautam Pathak

Dept./Lab OOPS Class 4B-1 Expt./No. 6

Title String Handling

6) Write a program to reverse words in a sentence without using library method.

```
import java.util.Scanner;

public class ReverseWords
{
  String str, str1;
  public ReverseWords (String str)
  {
    this.str = str;
    this.str = this.str + " ";
    this.str1 = " ";
  }
 Void reverse()
 {
  String temp = " ";
  int length = Str.length();
```

Signature of Teacher incharge

```
for(int i=0; i<length; i++)
{
    char ch = str.charAt(i);
    if(ch != ' ')
    {
        temp = Character.toString(ch) + temp;
    }
    if(ch == ' ')
    {
        str1 = str1 + temp;
        str += ch;
        temp = "";
    }
}
System.out.println(str1);
}

public static void main(String[] args)
{
    String str_input;
    Scanner s1 = new Scanner(System.in);
    System.out.println("Enter string :");
    str_input = s1.nextLine();
    ReverseWords RV = new ReverseWords(str_input);
```

Title String Handling

RV. reverse ( ) =

{

}

O/P

Enter string :

gautam

Reversed string :

mataug

Enter string :

RV College

Reversed Strng :

VR egelloC

Date 22/9/22

Dept./Lab OOPS Class 4B-1 Expt./No. 7

Title Collections

7) Write a program to create list containing, n copies of Specified Object Example.

Employee.java →

```
public class Employee
{
 String name, id, salary;
 Employee ( String name, String id, String salary)
 {
   this.id = id;
   this.name = name;
   this.salary = salary;
 }

 public String toString ()
  {
   return "Employee\n id= "
          + id + "\n Name = " + name +
           "\n Salary = " + salary;
```

Signature of Teacher incharge

List_Collection.java →

```
import java.util.*;

public class List_Collection
{
 public static void main(String[] args)
  {
   String id, name, sal;
   Scanner s = new Scanner ( System.in);
   System.out.println("Enter id : ");
   id = s.nextLine();
   System.out.println("Enter Name : ");
   name = s.nextLine();
   System.out.println("Enter Salary : ");
   sal = s.nextLine();

   Employee E = new Employee (id, name, sal);
   List<Employee> empCopies = Collections.nCopies(5, E);

   Iterator itr = empCopies.iterator();
   while( itr.hasNext())
   System.out.println(itr.next());
}

}
```