ASSIGNMENT 5

Design a class to implement stack concept with all necessary methods.

CODE:

Stack.java:

```
package com.company;

import java.util.Scanner;

public class Stack
{
    int top = -1, size;
    int[] a;
    Scanner scanner = new Scanner(System.in);
    void createarray()
    {
        System.out.println("Enter the size of the array");
        this.size = scanner.nextInt();
        a = new int[size];
    }
    void pop()
    {
        if (top == -1)
        {
            System.out.println("The stack is empty.");
        }
        else
        {
            top--;
        }
    }
    void push()
    {
        if (top == size - 1)
        {
            System.out.println("The stack is full.");
        }
        else
        {
            int element;
            System.out.println("Enter the element to insert in
the array");
            element = scanner.nextInt();
            a[++top] = element;
        }
    }
```

```
    void display()
    {
        if (top == -1)
        {
            System.out.println("No elements in the array
present");
        }
        else
        {
            System.out.println("Displaying the stack
elements");
            for (int i = 0; i <= top; i++)
            {
                System.out.println(a[i]);
            }
        }
    }
}
```

Main.java:

```
package com.company;

import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        Stack stack = new Stack();
        Scanner scanner = new Scanner(System.in);
        int choice = 0;
        while (choice != 5)
        {
            System.out.println("Press 1.CreateStack 2.Push
3.Pop 4.Display 5.Exit");
            choice = scanner.nextInt();
            switch (choice)
            {
                case 1:
                    stack.createarray();
                    System.out.println("The stack is
created");
                    break;
                case 2:
                    stack.push();
                    break;
                case 3:
                    stack.pop();
                    break;
                case 4:
                    stack.display();
```

```
                    break;
                case 5:
                    System.out.println("Thank you");
                    break;
                default:
                    System.out.println("Please enter valid
choice");
            }
        }
        scanner.close();
    }
}
```

OUTPUT:

```
Press 1.CreateStack 2.Push 3.Pop 4.Display 5.Exit
1
Enter the size of the array
3
The stack is created
Press 1.CreateStack 2.Push 3.Pop 4.Display 5.Exit
2
Enter the element to insert in the array
55
Press 1.CreateStack 2.Push 3.Pop 4.Display 5.Exit
2
Enter the element to insert in the array
7
Press 1.CreateStack 2.Push 3.Pop 4.Display 5.Exit
2
Enter the element to insert in the array
99
Press 1.CreateStack 2.Push 3.Pop 4.Display 5.Exit
4
Displaying the stack elements
55
7
99
```

Assignment 6

Q1.) Take a string from keyboard and convert into character array (new one).

CODE:

```
package package com.company;

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("Please enter a string : ");
        String str = s.next();
        char[] str1 = new char[str.length()];
        for (int i = 0; i < str.length(); i++) {
            str1[i] = str.charAt(i);
        }
        System.out.println("The new character array : \n");
        for (int i = 0; i < str.length(); i++) {
            System.out.println(i + " : " + str1[i]);
        }
    }
}
```

OUTPUT:

```
Please enter a string : parijat
The new character array :

0 : p
1 : a
2 : r
3 : i
4 : j
5 : a
6 : t
```

Q2) Take a string from keyboard and a char array (filled up to length 5). Now append the string to that char array. Show the char array.

CODE:

```
package com.company;
import java.util.*;
public class Main
{
    public static void main(String[] args)
    {
        Scanner scanner =new Scanner(System.in);
        String str;
        System.out.println("Enter a string");
        str=scanner.next();
        System.out.println("Enter the character array");
        char[] a=new char[5];
        for(int i=0;i<5;i++)
        {
            a[i]=scanner.next().charAt(0);
        }
        System.out.println("String after concatenating the
array");
        String s=" ";
        for(int i=0;i<5;i++)
        {
            s=s+a[i];
        }
        System.out.print(s+" "+str);
        scanner.close();
    }
}
```

OUTPUT:

```
Enter a string
Parijat
Enter the character array
H
e
l
l
o
String after concatenating the array
 Hello Parijat
Process finished with exit code 0
```

Q3) Find length of a string taken from keyboard and also find the length of that string except front and end spaces.

CODE:

```
package com.company;
import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Please enter a string : ");
        String str = s.nextLine();
        System.out.println("Length of the string : " +
str.length());
        System.out.println("Length of the string without
spaces : " + str.trim().length());
    }
}
```

OUTPUT:

```
Please enter a string :
Parijat
Length of the string : 7
Length of the string without spaces : 7

Process finished with exit code 0
```

Q4) Check if "Tech" presents in "University of Technology" or not. If yes return its position.

CODE:

```
package com.company;
public class Main {
    public static void main(String[] args) {
        String mainWord = "University of Technology";
        String toSearch = "Tech";
        if (mainWord.contains(toSearch)) {
            System.out.println("Found at position : " +
                    mainWord.indexOf(toSearch));
        }
```

```
        else {
            System.out.println("Word not found");
        }
    }
}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-18\bin\j
Found at position : 14

Process finished with exit code 0
```

Q5) Write a program to take a sentence and convert it into string arrays and sort the words using any sorting technique.

CODE:

```
package com.company;
import java.util.Scanner;
import java.util.Arrays;
public class Main{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter a sentence(max 10 words) :
");
        String line = s.nextLine();
        String[] words = line.split(" ", 10);
        Arrays.sort(words);
        System.out.println("Words in sorted order:");
        for (int i = 0; i < words.length; i++) {
            System.out.println(i + " : " + words[i]);
        }
    }
}
```

OUTPUT:

```
Enter a sentence(max 10 words) :
Hello myself Parijat
Words in sorted order:
0 : Hello
1 : Parijat
2 : myself

Process finished with exit code 0
```

Q6) Generate password from initials of one's first_name, middle_name, last_name and with last four digit of your roll_no (if middle name is not present, it won't come).

CODE:

```
package com.company;
import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter your full name : ");
        String temp = s.nextLine();
        String[] name = temp.split(" ", 3);
        System.out.println("Enter your roll");
        String roll = s.next();
        if (name.length != 3 || roll.length() < 4) {
            System.out.println("Password can't be generated");
        }
        else {
            String password = "";
            password += name[0].charAt(0);
            password += name[1].charAt(0);
            password += name[2].charAt(0);
            password += roll.substring(roll.length() - 4);
            System.out.println("Password : " + password);
        }
    }
}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-18\bin\
Enter your full name :
Sachin Ramesh Tedulkar
Enter your roll
18426
Password : SRT8426

Process finished with exit code 0
```

Q7) Write a program in Java which will read a string and rewrite it in the alphabetical order. For example, the word STRING should be written as GINRST.

CODE:

```
package com.company;
import java.util.*;
public class Main{
    public static void main(String[] args)
    {
        Scanner scanner=new Scanner(System.in);
        String str;
        System.out.println("Enter a string you want");
        str=scanner.next();
        char[] s=new char[str.length()];
        for(int i=0;i<str.length();i++)
        {
            s[i]=str.charAt(i);
        }
        Arrays.sort(s);
        System.out.println("The sorted array is");
        for(int i=0;i<str.length();i++)
        {
            System.out.print(s[i]);
        }
        scanner.close();
    }
}
```

OUTPUT:

```
Enter a string you want
parijat
The sorted array is
aaijprt
Process finished with exit code 0
```

Q8) Write a program in Java to extract a portion of a character string and print the extracted string. Assume that m characters are extracted, starting with the n-th character. The method signature will be like: void extract(String str, int n, int m).

CODE:

```
package com.company;
import java.util.*;
public class Main{
    public static void extract(String str,int n,int m)
    {
        System.out.println("The extracted substring is : ");
        for(int i=n-1;i<=(m+n-2);i++)
        {
            System.out.print(str.charAt(i));
        }
    }
    public static void main(String[] args)
    {
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter a string");
        String str=scanner.next();
        System.out.println("Enter the value of n");
        int n=scanner.nextInt();
        System.out.println("Enter the value of m");
        int m=scanner.nextInt();
        extract(str, n, m);
        scanner.close();
    }
}
```

OUTPUT:

```
C:\Users\A505\.jdks\openjdk-18\bin\
Enter a string
Ambitions
Enter the value of n
1
Enter the value of m
8
The extracted substring is :
Ambition
Process finished with exit code 0
```

Assignment -7

1.Write a program to handle the ArithmeticException.

CODE:

```
package com.company;
import java.util.Scanner;
public class Main {
    public static void main(String[] args)
    {
        try
        {
            Scanner scanner=new Scanner(System.in);
            System.out.println("Enter the Dividend");
            int x=scanner.nextInt();
            System.out.println("Enter the divisor");
            int y=scanner.nextInt();
            int z=x/y;
            System.out.println("The Quotient is "+z);
            scanner.close();
        }
        catch(ArithmeticException e)
        {
            System.out.println("We can not divide by 0");
        }
    }
}
```

OUTPUT:

```
Enter the Dividend
36
Enter the divisor
6
The Quotient is 6


Process finished with exit code 0
```

2.Write a program for multiple catch to fire ArrayIndexOutOfBoundsException and StringIndexOutOfBoundsException both.

CODE:

import java.util.Scanner;

```
package com.company;
import java.util.Scanner;
public class Ex {
    public static void main(String[] args)
    {
        try
        {
            Scanner scanner=new Scanner(System.in);
            int n;
            System.out.println("Enter the array size");
            n=scanner.nextInt();
            int[] arr=new int[n];
            System.out.println("Enter the array elements");
            for(int i=0;i<=n;i++)
            {
                int x;
                x=scanner.nextInt();
                arr[i]=x;
            }
            scanner.close();
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("The array index is out od
bound");
        }
    }
}
```

String code:

```
package com.company;
import java.util.Scanner;
public class Ex {
    public static void main(String[] args)
    {
        try
        {
            Scanner scanner=new Scanner(System.in);
            int n;
            System.out.println("Enter the string size");
            n=scanner.nextInt();
            String x="";
            for(int i=0;i<n;i++)
            {
```

```
                char a;
                a=scanner.next().charAt(0);
                x=x+a;

            }
            System.out.println("Displaying the string");
            System.out.println();
            for(int i=0;i<=x.length();i++)
            {
                System.out.print(x.charAt(i));
            }
            System.out.println();
            scanner.close();
        }
        catch(StringIndexOutOfBoundsException e)
        {
            System.out.println("You are accessing string index
as out of bound");
        }
    }
}
```

OUTPUT:

```
PS E:\javaprograms\hjk> cd "e:\javaprograms\hjk\" ; if ($?) { javac Ex.java }
if ($?) { java Ex }
Enter the array size
5
Enter the array elements
10
11
12
13
14
15
The array index is out of bound
```

```
if ($?) { java Ex }
Enter the string size
5
D
e
b
a
s
Displaying the string
DebasYou are accessing string index as out of bound
```

3.Write a program to fire the NegativeArraySize exception.

CODE:

```
package com.company;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        try {
            Scanner scanner = new Scanner(System.in);
            int n;
            System.out.println("Enter the array size");
            n = scanner.nextInt();
```

```
            int[] a = new int[n];
            for (int i = 0; i < n; i++) {
                int x;
                x = scanner.nextInt();
                a[i] = x;
            }
            System.out.println("The array is: ");
            for (int i = 0; i < n; i++) {
                System.out.print(a[i]);
            }
            scanner.close();
        } catch (NegativeArraySizeException e) {
            System.out.println("The array can not be of
negative size");
        }
    }
}
```

OUTPUT:

```
Enter the array size
-2
The array can not be of negative size


Process finished with exit code 0
```

4. Define an object reference and initialize it to null. Try to call a method through this reference. Now wrap the in a try-catch clause to catch the exception.

CODE:

```
package com.company;
import java.util.Scanner;
class Cat
{
    void call()
    {
        System.out.println("Cat calls meow");
    }
}
public class Main {
    public static void main(String[] args)
    {
        try
        {
            Cat cat=null;
            cat.call();
```

```
        }
        catch(NullPointerException e)
        {
            System.out.println("The null pointer can not call
a method");
        }
    }
}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-18\bin\java.exe
The null pointer can not call a method

Process finished with exit code 0
```

5. Write a program in Java to create a user defined exception named PayOutOfBoundsException (provided the monthly salary of a person is less than Rs. 10,000 /) and fire the exception.

CODE:

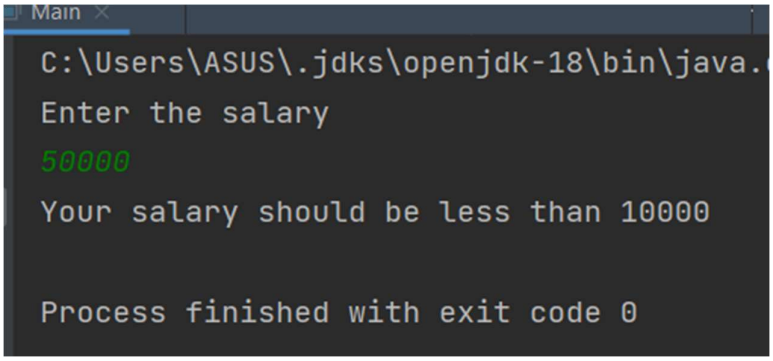
```
package com.company;
import java.util.Scanner;
class PayOutOfBounds extends Exception {
    public PayOutOfBounds(double salary)
    {

    }
}
public class Main {
    public static void main(String[] args)
    {

        try
        {
            double salary;
            Scanner scanner=new Scanner(System.in);
            System.out.println("Enter the salary");
            salary=scanner.nextDouble();
            if(salary>10000)
            {
                throw new PayOutOfBounds(salary);
            }
            System.out.println("Salary is "+salary);
            scanner.close();
        }
        catch(PayOutOfBounds e)
        {
```

```
            System.out.println("Your salary should be less
than 10000");
        }
    }
}
```

OUTPUT:



6. Write a program to fire any checked exception manually using 'throw' keyword.

CODE:

```
package com.company;
public class Main{
    static void fun()
    {
        try
        {
            throw new NullPointerException("demo");
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught inside fun().");
            throw e;
        }
    }
    public static void main(String args[])
    {
        try
        {
            fun();
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught in main.");
        }
    }
}
```

OUTPUT:

```
Caught inside fun().
Caught in main.

Process finished with exit code 0
```

7. Create a class with two methods, f( ) and g( ). In g( ), throw an exception of a new type that you define. In f( ), call g( ), catch its exception and, in the catch clause, throw a different exception (of a second type that you define). Test these methods from and within main( ).

CODE:

```
package com.company;
public class Main{
    static void f()
    {
        try
        {
            throw new NullPointerException("demo");
        }
        catch(NullPointerException e)
        {
            System.out.println("Caught inside f().");
            throw e;
        }
    }
    static void g(){
        try
        {
            throw new ArrayIndexOutOfBoundsException("demo2");

        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("Caught inside g().");
            throw e;
        }
    }
    public static void main(String args[]){
        try
        {
            f();
        }
        finally
        {
            try
```

```
            {
                g();
            }
            catch(NullPointerException e)
            {
                System.out.println("Caught in main.");
            }
            catch(ArrayIndexOutOfBoundsException f)
            {
                System.out.println("Caught in main.");
            }
        }
    }
}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-
Caught inside f().
Caught inside g().
Caught in main.
```

8. Write a program that takes one string and two integers as command line argument and prints the reverse of the substring of the string specified by the two numbers. The program should handle all possible exception that may arise due to bad input.

CODE:

```
package com.company;
public class Subclass{
    public static void main(String[] args)
    {
        try{
            String s=args[0];
            int b=Integer.parseInt(args[1]);
            int c=Integer.parseInt(args[2]);
            String str="";
            for(int i=b;i<=c;i++)
            {
                str=str+s.charAt(i);
            }
            String p="";
            for(int i=0;i<str.length();i++)
            {
                p=p+str.charAt(i);
            }
            System.out.println("The reversed substring is
"+p);
        }
```

```
        catch(Exception e)
        {
            System.out.println("Please check your inserted
string or given indexes");
        }
    }
}
```

OUTPUT:

```
PS C:\Users\Debashis\OneDrive\Desktop> javac SubClass.java
PS C:\Users\Debashis\OneDrive\Desktop> java SubClass Debashis 2 4
The reversed substring is bas
```

9. Write a demo program to illustrate the restrictions of using 'throws' clause in method overriding with regard to superclass-subclass concept.

CODE:

```
package com.company;
class SuperClass {
    void method() {
        System.out.println("SuperClass");
    }
}
class Main extends SuperClass {
    void method() throws ArithmeticException {

        System.out.println("SubClass");
    }
    public static void main(String args[]) {
        SuperClass s = new Main();
        s.method();
    }
}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-18\bin\
SubClass


Process finished with exit code 0
```

Assignment 8

1. Create a class and determine if method overloading holds good for return type of methods or not

CODE:

```
   package com.company;
public class Main {
    public int  sum(int a,int b)
    {
        return (a+b);
    }
    public double sum(double a,double b)
    {
        return (a+b);
    }
    public static void main(String[] args)
    {
        Main method=new Main();
        int c=method.sum(3,4);
        System.out.println("The result of summation is "+c);
        double d=method.sum(1.2,5.3);
        System.out.println("The result of summation is "+d);
    }
}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-18\bin
The result of summation is 7
The result of summation is 6.5

Process finished with exit code 0
```

Hence method overloading holds good for java.

2. Overload the constructors for classes Area and Volume of a rectangular figure and displayits area and volume. Area is the superclass and Volume is the subclass.

CODE:

AREA CODE

```
package com.company;
import java.util.Scanner;
public class Area {
    double length,breadth;
    void set_dim()
    {
        Scanner scanner=new Scanner(System.in);
```

```
        System.out.println("Enter the length of the rectangle");
        double length=scanner.nextDouble();
        System.out.println("Enter the breadth of the
rectangle");
        double breadth=scanner.nextDouble();
        this.length=length;
        this.breadth=breadth;
        scanner.close();
    }
    public double cal_area()
    {
        System.out.println("The area is
"+(this.length*this.breadth));
        return (this.length*this.breadth);
    }
}
```

VOLUME CODE:

```
package com.company;
import java.util.Scanner;
public class Volume extends Area{
    double height;
    public void set_height(double z)
    {

        this.height=z;
    }
    public double cal_volume(Volume vol,double height)
    {
        vol.set_dim();
        vol.set_height(height);
        return vol.height*vol.cal_area();
    }
    public static void main(String[] args)
    {
        Volume vol=new Volume();
        Scanner scanner=new Scanner(System.in);
        System.out.println("Enter the height");
        double height=scanner.nextDouble();
        System.out.println("The volume is
"+vol.cal_volume(vol,height));
        scanner.close();
    }

}
```

OUTPUT:

```
PS E:\javaprograms\dij> cd "e:\javaprograms\dij\" ; if ($?) { javac Volum
e.java } ; if ($?) { java Volume }
Enter the height
10
Enter the length of the rectangle
20
Enter the breadth of the rectangle
12
The area is 240.0
The volume is 2400.0
```

3. Create a class Employee is having instance variables *name* and *id*. Create its subclass named Scientist which has instance variables *no_of_publication* and *experience*. Now create its subclass, say DScientist which has instance variable award. Put a method like: public String toString(){ } in every class where you describe about the class and from main() method create object of each class and print each object.

CODE:

Employee class:

```
   public class DScientist extends Scientist {
    int award;
    DScientist(int no_of_publications,int expirience,String name
,int id,int award)
    {
        super(no_of_publications, expirience, name, id);
        this.award=award;
    }
    public String toString()
    {
        String str="The award count is"+this.award;
        return str;
    }
```

Scientist class:

```
public class Scientist extends Employee {
    int no_of_publications,expirience;
    Scientist(int no_of_publications,int expirience,String name
,int id)
    {
        super(name,id);
        this.no_of_publications=no_of_publications;
        this.expirience=expirience;
    }
    public String toString()
    {
        String str="No of publication in his name
is"+this.no_of_publications+" His expirience
is"+this.expirience;
        return str;
    }
}
```

Dscientist class:

```
public class DScientist extends Scientist {
    int award;
    DScientist(int no_of_publications,int expirience,String name
,int id,int award)
    {
        super(no_of_publications, expirience, name, id);
        this.award=award;
    }
```

```
    public String toString()
    {
        String str="The award count is"+this.award;
        return str;
    }
    public static void main(String[] args)
    {
        Employee employee=new Employee("Debashis", 0);
        String str=employee.toString();
        System.out.println(str);
        Scientist scientist=new Scientist(4, 3, "Debashis", 0);
        str=scientist.toString();
        System.out.println(str);
        DScientist dScientist=new DScientist(4, 3, "Debashis",
0, 2);
        str=dScientist.toString();
        System.out.println(str);
    }
}
```

OUTPUT:

```
PS E:\javaprograms\dij> cd "e:\javaprograms\dij\" ; if ($?) { javac DScie
ntist.java } ; if ($?) { java DScientist }
The employee name is Debashis.  His id is 0
No of publication in his name is4 His expirience is3
The award count is2
```

4. Create a class with a method void show() and make three subclasses of it and all subclasses have this show() method overridden and call those methods using their corresponding object references.

CODE:

```
class Show
{
    public void show()
    {
        System.out.println("I am showing my name");
    }
}
class Show1 extends Show {
    public void show()
    {
        System.out.println("We show our marks");
    }
}
class Show2 extends Show {
    public void show()
    {
        System.out.println("We will not see the show");
    }
}
class Show3 extends Show {
```

```
    public void show()
    {
        System.out.println("We show nothing");
    }
}

public class Main
{
    public static void main(String[] args)
    {
        Show show=new Show();
        show.show();
        Show1 show1=new Show1();
        show1.show();
        Show2 show2=new Show2();
        show2.show();
        Show3 show3=new Show3();
        show3.show();
    }
}
```

OUTPUT:

```
PS E:\javaprograms\dij> cd "e:\javaprograms\dij\" ; if ($?) { javac Main.
java } ; if ($?) { java Main }
I am showing my name
We show our marks
We will not see the show
We show nothing
```

5. Do the problem 4 using dynamic method dispatching.

CODE:

```
class A
{
    void show()
    {
        System.out.println("We show our marks");
    }
}
class B extends A
{
    void show()
    {
        System.out.println("We will not see the show");
    }
}
class C extends A
{
    void show()
    {
        System.out.println("We show nothing");
```

```
    }
}
public class Main
{
    public static void main(String args[])
    {
        A a = new A();
        B b = new B();
        C c = new C();
        A ref;
        ref = a;
        ref.show();
        ref = b;
        ref.show();
        ref = c;
        ref.show();
    }
}
```

OUTPUT:

```
PS E:\javaprograms\dij> cd "e:\javaprograms\dij\" ; if ($?) { javac Main.
java } ; if ($?) { java Main }
We show our marks
We will not see the show
We show nothing
```

6. Check without having any abstract method/s whether a class can be abstract; if so, then use thatconcrete method(s) from another class having main method.

CODE:

```
package com.company;
abstract class AbstractClass{

    public static void nonAbstractMethodOne(String p1,String
p2){
        String param = p1 +" " + p2;
        System.out.println(param);
    }

    public static void nonAbstractMethodTwo(String p){
        System.out.println("Value of param is "+p);
    }
}
public class Main
{
    public static void main(String[] args) {
        String str1="Parijat";
        String str2="Priyadarshi";
        AbstractClass.nonAbstractMethodTwo(str2);
        AbstractClass.nonAbstractMethodOne(str1,str2);
    }
}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-18\bin
Value of param is Priyadarshi
Parijat Priyadarshi

Process finished with exit code 0
```

7. Create a class Parent having instance variables id, name and address. Create a class ChildOnehaving instance variables id, name, address and marks. Also create another class ChildTwo withinstance variables id, name, address, qualification and salary. Within each class define your ownmethod to display values of these variables. Design the program using super call with proper parameter and use object of each class from main() to display their properties.

CODE:

```
package com.company;
class Parent
{
    int id;
    String name,address;
    Parent(int id,String name,String address)
    {
        this.id=id;
        this.name=name;
        this.address=address;
    }
    public void display()
    {
        System.out.println("The id is"+id+" name : "+name+"
address: "+address);
    }
}

class ChildOne extends Parent {

    int marks;
    ChildOne(int id,String name,String address,int marks)
    {
        super(id,name,address);
        this.marks=marks;
    }
    public void display()
    {
        System.out.println("The id is"+id+" name : "+name+"
address: "+address+" marks: "+marks);
```

```
    }


}
class ChildTwo extends Parent {
    int salary;
    String qualification;
    ChildTwo(int id,String name,String address,String
qualification,int salary)
    {
        super(id,name,address);
        this.qualification=qualification;
        this.salary=salary;
    }
    public void display()
    {
        System.out.println("The id is"+id+" name : "+name+"
address: "+address+" qualification: "+qualification+" salary:
"+salary);
    }

}
public class Main {
    public static void main(String[] args)
    {
        Parent parent=new Parent(0,"Parijat","Salt LAke");
        parent.display();
        ChildOne child1=new ChildOne(1, "Parijat", "Kolkata",
100);
        child1.display();
        ChildTwo child2=new ChildTwo(2, "Rishu", "Ranchi",
"12th", 10000);
        child2.display();
    }

}
```

OUTPUT:

```
C:\Users\ASUS\.jdks\openjdk-18\bin\java.exe "-javaagent:C:\Program Files\Je
The id is0 name : Parijat address: Salt LAke
The id is1 name : Parijat address: Kolkata marks: 100
The id is2 name : Rishu address: Ranchi qualification: 12th salary: 10000

Process finished with exit code 0
```