Lab Cycle 5
Experiment No : 35
Date : 6/06/2022

AIM : Write a user defined exception class to authenticate the username and password.

ALGORITHM :
Step 1 : Start
Step 2 : Read username and password
Step 3 : Find the length of username
Step 4 : In try block define cases where exceptions may arise
    Step 4.1 : if username length is less than 6
        call UsernameException which inherits class Exception
        by passing message “Username must be greater than 6 characters”.
    Step 4.2 : else if password not equal to predifined string
        call PasswordException which inherits class Exception
        by passing message “Incorrect password Type correct password”
    Step 4.3 : else print “Login Successful !!”
Step 5 : if there is any of the above Userdefined Exception catch the exception and print the corresponding message.
Step 6 : Stop

SOURCE CODE :

```
import java.util.Scanner;
class UsernameException extends Exception{
    public UsernameException(String msg){
        super(msg);
    }
}
class PasswordException extends Exception{
    public PasswordException(String msg){
        super(msg);
    }
}
public class LoginException{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        String username,password;
        System.out.print("Enter username: ");
        username = s.nextLine();
        System.out.print("Enter password: ");
        password = s.nextLine();
        int length = username.length();
        try{
            if(length < 6){
```

```
                    throw new UsernameException("Username must be
                    greater than 6 characters ???>");}
                else if(!password.equals("abc@123")){
                    throw new PasswordException("Incorrect password\nType
                    correct password ???");}
                else
                    System.out.println("Login Successful !!!");
            }
            catch (UsernameException e){
                System.out.println(e);
            }
            catch (PasswordException p){
                System.out.println(p);
            }
        }
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac LoginException.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris
Enter password: chris
UsernameException: Username must be greater than 6 characters ???>
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris william
Enter password: chris
PasswordException: Incorrect password
Type correct password ???
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java LoginException
Enter username: chris william
Enter password: abc@123
Login Successful !!!
```

RESULT :

Program is successfully executed and output is verified.

Experiment No : 36
Date : 6/06/2022

AIM : Find the average of N positive integers, raising a user defined exception for each negative input.

ALGORITHM :

Step 1 : Start
Step 2 : set sum=0
Step 3 : Read the number of integers(n) to be read and initialize array arr[] of size n
Step 4 : set i=0
    Step 4.1 : for i<n read interger in
    Step 4.2 : if in< 0 then,
        call InputException which inherits class Exception
        by passing message “Number is not positive, type a positive number”.
    Step 4.3 : else set arr[i] = in
            set sum= sum+ arr[i]
    Step 4.4 : i=i+1
Step 5 : set avg=sum/n
Step 6 : print “average=avg”
Step 7 : if the read integer is negative catch the InputException and print the message.
Step 8 : Stop

SOURCE CODE :

```
import java.util.Scanner;
class InputException extends Exception{
    public InputException(String msg){
        super(msg);
    }
}
public class AverageException{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        int sum=0;
        System.out.print("Enter no of integers: ");
        int n = s.nextInt();
        int arr[]= new int[n];
        System.out.print("Enter the integers: ");
        try {
            for(int i=0;i<n;i++){
                int in=s.nextInt();
                if(in < 0){
                    throw new InputException("Number is not
                    positive\ntype a positive number ???");}
```

```
                    arr[i] = in;
                    sum += arr[i];
                }
                float avg= sum / n;
                System.out.println("Average is: "+avg);

            }
            catch (InputException e){
                System.out.println(e);
            }
        }
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac AverageException.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java AverageException
Enter no of integers: 3
Enter the integers: 15 -6 4
InputException: Number is not positive
type a positive number ???
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java AverageException
Enter no of integers: 3
Enter the integers: 15 6 4
Average is: 8.0
```

RESULT :

Program is successfully executed and output is verified.

Experiment No : 37
Date : 08/06/2022

AIM : Define two classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

ALGORITHM :

Step 1 : Start
Step 2 : create an object a for class multiplication which inherits Thread class
Step 3 : create an object b for class primenumbers which inherits Thread class
Step 4 : call run() in class multiplication by a.start()
    Step 4.1 : set i=1
    Step 4.2 : if i<=10
        print “i x 5 = i*5”
        i = i+1 go to step 4.2
    Step 4.3 : else exit from multiplication thread
Step 5 : call run() in class primenumbers by b.start()
    Step 5.1 : Read the limit value n
    Step 5.2 : call prime_N()
        Step 5.2.1 : set x=1
        Step 5.2.2 : if x <= n do step 4.2.3 to 4.2.10
        Step 5.2.3 : check if x=1 or x=0 then i=i+1
        Step 5.2.4 : set flag=1
        Step 5.2.5 : set y=2
        Step 5.2.6 : if y <= x/2 do steps 4.2.7 to 4.2.8
        Step 5.2.7 : if x % y = 0 set flag=0 and go to 4.2.9
        Step 5.2.8 : y=y+1 go to step 4.2.6
        Step 5.2.9 : if flag = 1 print x
        Step 5.2.10 : print new line go to step 4.2.2
Step 6 : Stop

SOURCE CODE :

```
import java.util.*;
class multiplication extends Thread{
    public void run(){
        System.out.println("Multiplication table of 5 : ");
        for(int i = 1; i <= 10; i++){
            System.out.println(i + " X 5 =" + i*5 +"\n");}
        System.out.println("Exiting from Thread Multiplication ...");
    }
}
```

```
class primenumbers extends Thread{
	public void run(){
		Scanner sc=new Scanner(System.in);
		System.out.print("Enter the value of n: ");
		int n=sc.nextInt();
		prime_N(n);
	}
	static void prime_N(int N){
		int x,y,flag;
		System.out.println("All the Prime numbers within 1 and " + N + " are:");
		for (x = 1; x <= N; x++){
		if (x == 1 || x == 0)
				continue;
		flag = 1;
		for (y = 2; y <= x / 2; ++y){
				if (x % y == 0) {
				flag = 0;
				break;
				}
		}
		if (flag == 1)
				System.out.print( x + "\t");
		}
		System.out.println();
	}
}
public class ThreadClass{
	public static void main(String args[]){
		multiplication a = new multiplication();
		primenumbers b = new primenumbers();
		a.start();
		b.start();
	}
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac ThreadClass.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java ThreadClass
Multiplication table of 5 :
1 X 5 =5

2 X 5 =10

3 X 5 =15

4 X 5 =20

5 X 5 =25

6 X 5 =30

7 X 5 =35

8 X 5 =40

9 X 5 =45

10 X 5 =50

Exiting from Thread Multiplication ...
Enter the value of n: 15
All the Prime numbers within 1 and 15 are:
2       3       5       7       11      13
```

RESULT :

Program is successfully executed and output is verified.

Experiment No : 38
Date : 08/06/2022

AIM : Define two classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

ALGORITHM :
Step 1 : Start
Step 2 : Read the range limit n
Step 3 : create an object a of class Even which Inherit Runnable class by passing the value of n. this intializes value of n
Step 4 : create object t1 for Thread class by passing object a of Even class
Step 5 : call run() function of Runnable class using object t1
t1.start()
Step 6 : in run() set i=1
Step 6.1 : if i<n
check if i%2==0 then print “i is an even number”
Step 7 : create an object b of class Fibonnaci which Inherit Runnable class by passing the value of n. this intializes value of n
Step 8 : create object t2 for Thread class by passing object b of Fibonnaci class
Step 9 : call run() function of Runnable class using object t2
t2.start()
Step 10 : in run() set n=a
a=0,b=1
Step 10.1 : if n=0 print “term 0 : a”
Step 10.2 : else if n=1 print “term 0 : a \n term 1 : b”
Step 10.3 :else print “term 0 : a \n term 1 : b”
set i=2
for i <=n
set c= a+b
set a=b
set b=c
print “\nterm i : b”
Step 11 : Stop

SOURCE CODE :

```
import java.util.*;
class Fibonnaci implements Runnable{
        int a;
        Fibonnaci(int a){
                this.a=a;
        }
        public void run(){
                int n=a;
                int a = 0, b = 1, c;
```

```
                if (n == 0)
                System.out.println("term 0 : " + a);
                else if (n ==1)
                System.out.println("term 0 : " + a + "\nterm 1: "+ b);
                else{
                        System.out.println("term 0 : " + a + "\nterm 1: "+ b);
                        for (int i = 2; i <= n; i++){
                        c = a + b;
                        a = b;
                        b = c;
                                System.out.println("term " + i + ": " + b);
                        }
                }
                System.out.println();
        }
}
class Even implements Runnable{
        int b;
        Even(int b){
                this.b=b;
        }
        public void run(){
                for(int i=1;i<=b;i++){
                        if(i%2 == 0)
                                System.out.println(i + " " + "is an even number");
                }
                System.out.println("\n Fibonacci series: ");
        }
}
public class RunnableClass{
        public static void main(String args[]){
                Scanner sc=new Scanner(System.in);
                System.out.print("Enter the range limit: ");
                int n=sc.nextInt();
                Even a = new Even(n);
                Thread t1= new Thread(a);
                t1.start();
                Fibonnaci b = new Fibonnaci(n);
                Thread t2= new Thread(b);
                t2.start();
        }
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac RunnableClass.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java RunnableClass
Enter the range limit: 7
2 is an even number
4 is an even number
6 is an even number

 Fibonacci series:
term 0 : 0
term 1: 1
term 2: 1
term 3: 2
term 4: 3
term 5: 5
term 6: 8
term 7: 13
```

RESULT :

Program is successfully executed and output is verified.

Experiment No : 39
Date : 10/06/2022

AIM : Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

ALGORITHM :

Step 1 : Start
Step 2 : create a package Graphics
Step 3 : create 4 source files with interfaces and classes to find area of figures Rectangle, Triangle, Square and Circle.
Step 4 : write package name at the top of every source file
Step 5 : Create a main program that imports the package Graphics with classes Rectangle, Triangle, Square and Circle .
Step 6 : read the parameters of figures (Rectangle, Triangle, Square and Circle) and create an object to invoke the methods defined in the classes of package inorder to find the area
Step 7 : Stop

SOURCE CODE :

Prgrm5.java

```
import java.util.Scanner;
import Graphics.Rectangle;
import Graphics.Triangle;
import Graphics.Square;
import Graphics.Circle;
public class Prgrm5{
	public static void main(String args[]){
		Scanner sc=new Scanner(System.in);
		int i=1,ch;
		System.out.println("_____Program to find areas of different shapes_____");
		while(i!=0){
			System.out.println("1. Rectangle \t2. Square \t3. Circle \t4. Triangle");
			System.out.print("Choose a option to find the area: ");
			ch=sc.nextInt();
			switch(ch){
				case 1: System.out.println("Enter the length and breadth of rectangle: ");
					float len=sc.nextFloat();
					float bdth=sc.nextFloat();
					Rectangle obj1=new Rectangle();
```

```
                                        obj1.setdata(len,bdth);
                                        obj1.area();
                                        break;
                                case 2: System.out.println("Enter the side length of the
square: ");
                                        float side=sc.nextFloat();
                                        Square obj2=new Square();
                                        obj2.setdata(side);
                                        obj2.area();
                                        break;
                                case 3: System.out.println("Enter the radius of circle: ");
                                        float rad=sc.nextFloat();
                                        Circle obj3=new Circle();
                                        obj3.setdata(rad);
                                        obj3.area();
                                        break;
                                case 4: System.out.println("Enter the 3 sides of triangle :
");
                                        float a=sc.nextFloat();
                                        float b=sc.nextFloat();
                                        float c=sc.nextFloat();
                                        Triangle obj4=new Triangle();
                                        obj4.setdata(a,b,c);
                                        obj4.area();
                                        break;
                        }
                        System.out.print("Do you want to continue?(0(no),1(yes)) ");
                        i=sc.nextInt();
                }
        }
}
```

Rectangle.java

```
package Graphics;
interface interface1{
        void area();
}
public class Rectangle implements interface1{
        float length,breadth;
        public void setdata(float len,float bdth){
                length=len;breadth=bdth;
        }
        public void area(){
                System.out.println("Area of rectangle is : " + (length*breadth));
        }
```

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

Square.java

```
package Graphics;
interface interface1{
	void area();
}
public class Square implements interface1{
	float side;
	public void setdata(float side){
		this.side=side;
	}
	public void area(){
		System.out.println("Area of Square is : " + (side*side));
	}
public static void main(String args[]){}
}
```

Triangle.java

```
package Graphics;
interface interface1{
	void area();
}
public class Triangle implements interface1{
	float a,b,c;
	public void setdata(float a,float b,float c){
		this.a=a;this.b=b;this.c=c;
	}
	public void area(){
		float s=(a+b+c)/2;
		double a_rea=Math.sqrt(s*(s-a)*(s-b)*(s-c));
		System.out.println("Area of triangle is : " + a_rea);
	}
public static void main(String args[]){}
}
```

Circle.java

```
package Graphics;
interface interface1{
	void area();
}
public class Circle implements interface1{
	float radius;
	public void setdata(float rad){
```

```
            radius=rad;
        }
        public void area(){
            System.out.println("Area of circle is : " + (3.14*radius*radius));
        }

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

}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Rectangle.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Graphics.Rectangle
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Square.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Graphics.Square
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Triangle.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Graphics.Triangle
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Circle.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Graphics.Circle
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Prgrm5.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Prgrm5
___________Program to find areas of different shapes___________
1. Rectangle    2. Square       3. Circle       4. Triangle
Choose a option to find the area: 1
Enter the length and breadth of rectangle:
6
6
Area of rectangle is : 36.0
Do you want to continue?(0(no),1(yes)) 1
1. Rectangle    2. Square       3. Circle       4. Triangle
Choose a option to find the area: 2
Enter the side length of the square:
4
Area of Square is : 16.0
Do you want to continue?(0(no),1(yes)) 1
1. Rectangle    2. Square       3. Circle       4. Triangle
Choose a option to find the area: 3
Enter the radius of circle:
6.4
Area of circle is : 128.61440383300783
Do you want to continue?(0(no),1(yes)) 1
1. Rectangle    2. Square       3. Circle       4. Triangle
Choose a option to find the area: 4
Enter the 3 sides of triangle :
3.4
5.2
3.3
Area of triangle is : 5.4913904254689125
Do you want to continue?(0(no),1(yes)) 0
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$
```

RESULT :

Program is successfully executed and output is verified.

Experiment No : 40
Date : 13/06/2022

AIM : Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

ALGORITHM :

Step 1 : Start
Step 2 : create a package Arithmetic
Step 3 : create 4 source files with interfaces and classes to implement Arithematic operations Addition,Subtraction,Multiplication,Division
Step 4 : write package name at the top of every source file
Step 5 : Create a main program that imports the package Arithmetic with classes Addition,Subtraction,Multiplication,Division.
Step 6 : read the numbers to do Arithematic operation (Addition, Subtraction, Multiplication, Division) and create an object to invoke the methods defined in the classes of package inorder to do the operations
Step 7 : Stop

SOURCE CODE :

Prgrm6.java

```
import java.util.Scanner;
import Arithmetic.Addition;
import Arithmetic.Subtraction;
import Arithmetic.Multiplication;
import Arithmetic.Division;
public class Prgrm6{
	public static void main(String args[]){
		Scanner sc=new Scanner(System.in);
		int i=1,ch;
		System.out.println("_____Program to demonstrate the basic arithmetic operations_____");
		System.out.println("Enter two numbers: ");
		float n1=sc.nextFloat();
		float n2=sc.nextFloat();
		while(i!=0){
			System.out.println("1. Addition \t2. Subtraction \t3. Multiplication \t4. Division");
			System.out.print("Choose an option: ");
			ch=sc.nextInt();
			switch(ch){
				case 1: Addition obj1=new Addition();
```

```
                                        obj1.calculate(n1,n2);
                                        break;
                                case 2: Subtraction obj2=new Subtraction();
                                        obj2.calculate(n1,n2);
                                        break;
                                case 3: Multiplication obj3=new Multiplication();
                                        obj3.calculate(n1,n2);
                                        break;
                                case 4: Division obj4=new Division();
                                        obj4.calculate(n1,n2);
                                        break;
                        }
                        System.out.print("Do you want to continue?(0(no),1(yes)) ");
                        i=sc.nextInt();
                }

        }
}
```

Addition.java

```
package Arithmetic;
interface interface1{
        void calculate(float a,float b);
}
public class Addition implements interface1{
        public void calculate(float a,float b){
                System.out.println("Sum of the given numbers is : " + (a+b));
        }
public static void main(String args[]){}
}
```

Subtraction.java

```
package Arithmetic;
interface interface1{
        void calculate(float a,float b);
}
public class Subtraction implements interface1 {
        public void calculate(float a,float b){
                System.out.println("Difference of the given numbers is : " + (a-b));
        }
public static void main(String args[]){}
}
```

Multiplication.java

```
package Arithmetic;
interface interface1{
        void calculate(float a,float b);
}
```

```
public class Multiplication implements interface1{
        public void calculate(float a,float b){
                System.out.println("Product of the given numbers is : " + (a*b));
        }
public static void main(String args[]){}
}
```

Division.java

```
package Arithmetic;
interface interface1{
        void calculate(float a,float b);
}
public class Division implements interface1{
        public void calculate(float a,float b){
                System.out.println("Quotient of the given numbers is : " + (a/b));
        }
public static void main(String args[]){}
}
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Addition.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Addition
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Subtraction.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Subtraction
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Multiplication.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Multiplication
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac -d . Division.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Arithmetic.Division
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Prgrm6.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Prgrm6
_____Program to demonstrate the basic arithmetic operations_____
Enter two numbers:
8
4.2
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 1
Sum of the given numbers is : 12.2
Do you want to continue?(0(no),1(yes)) 1
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 2
Difference of the given numbers is : 3.8000002
Do you want to continue?(0(no),1(yes)) 1
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 3
Product of the given numbers is : 33.6
Do you want to continue?(0(no),1(yes)) 1
1. Addition     2. Subtraction  3. Multiplication       4. Division
Choose an option: 4
Quotient of the given numbers is : 1.904762
Do you want to continue?(0(no),1(yes)) 0
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$
```

RESULT :

Program is successfully executed and output is verified.

Experiment No : 41
Date : 15/06/2022

AIM : Program to create a generic stack and do the Push and Pop operations.

ALGORITHM :

Step 1 : Start
Step 2 : create a generic class for stack and declare a Arraylist(arr) ,define methods push() and pop()
Step 3 : Read the size of stack
Step 4 : Prompt user to choose integer or string type of generic stack
Step 5 : if choosed option is integer create an integer object and do any of the following operations
Step 5.1 : to push : read an integer and invoke push() with value as argument
Step 5.2 : to pop : invoke pop ()
Step 6 : else if choosed option is string create an string object and do any of the following operations
Step 6.1 : to push : read the string and invoke push() with string as argument
Step 6.2 : to pop : invoke pop ()
Step 7 : Stop

SOURCE CODE :

```
import java.util.*;
class Stack<T>{
        private int top,size;
        ArrayList<T> arr;

        Stack(int data1){
                size = data1;
                top=-1;
                arr=new ArrayList<T>(size);
        }
        public void push(T data){
                if(top == size - 1){
                        System.out.println("Stack OverFlow");
                        System.exit(1);
                }
                top=top+1;
                arr.add(data);
        }
        public void pop(){
                if (top == -1){
                        System.out.println("Stack Underflow");
                        System.exit(1);  }
                arr.remove(top);
```

```
                top=top-1;      }
        public void printstack(){
                System.out.print("Stack elements are: ");
                Iterator<T> iterator = arr.iterator();
                while(iterator.hasNext()){
                                System.out.print(iterator.next() + "\t");       }
                                System.out.println();           }
}
class Prgrm7{
        public static void main(String[] args){
                Scanner sc=new Scanner(System.in);
                Scanner s=new Scanner(System.in);
                int i=1,j=1;
                System.out.print("Enter the size of the stack: ");
                int size=sc.nextInt();
                while(j!=0){
                int ch1,ch2;
                i=j=1;
                System.out.println("1.Generic stack as integer type \t2.Generic stack as
string type");
                System.out.print("Choose an option: ");
                ch1=sc.nextInt();
                if(ch1 == 1){
                        Stack<Integer> intObj = new Stack<>(size);
                        while(i!=0){
                                System.out.println("Generic stack as integer type!!!");
                                System.out.println("1. Push \t2. Pop ");
                                System.out.print("Choose an option: ");
                                ch2=sc.nextInt();
                                switch(ch2){
                                        case 1: System.out.print("Enter element to be
added : ");
                                                intObj.push(sc.nextInt());
                                                intObj.printstack();
                                                break;
                                        case 2: intObj.pop();
                                                intObj.printstack();
                                                break;     }
        System.out.print("Do you want to continue?(0(no),1(yes)) ");
                                i=sc.nextInt();
                        } }
                else if(ch1 == 2){
                        Stack<String> stringObj = new Stack<>(size);
                        while(i!=0){
                                System.out.println("Generic stack as string type!!!");
                                System.out.println("1. Push \t2. Pop ");
```

```
                        System.out.print("Choose an option: ");
                        ch2=sc.nextInt();
                        switch(ch2){
                                case 1: System.out.print("Enter a string: ");
                                        stringObj.push(s.nextLine());
                                        stringObj.printstack();
                                        break;
                                case 2: stringObj.pop();
                                        stringObj.printstack();
                                        break;
                        }
                System.out.print("Do you want to continue?(0(no),1(yes)) ");
                        i=sc.nextInt();
                }}
        else
                System.out.println("Invalid option");
        System.out.print("continue?(0(no),1(yes)) ");
        j=sc.nextInt();
        }
} }
```

OUTPUT :

```
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ javac Prgrm7.java
lab@lab-Lenovo-IdeaPad-Z400:~/Documents/labcycle4(java)$ java Prgrm7
Enter the size of the stack: 4
1.Generic stack as integer type         2.Generic stack as string type
Choose an option: 1
Generic stack as integer type!!!
1. Push         2. Pop
Choose an option: 1
Enter element to be added : 2
Stack elements are: 2
Do you want to continue?(0(no),1(yes)) 1
Generic stack as integer type!!!
1. Push         2. Pop
Choose an option: 1
Enter element to be added : 3
Stack elements are: 2   3
Do you want to continue?(0(no),1(yes)) 1
Generic stack as integer type!!!
1. Push         2. Pop
Choose an option: 1
Enter element to be added : 6
Stack elements are: 2   3       6
Do you want to continue?(0(no),1(yes)) 1
Generic stack as integer type!!!
1. Push         2. Pop
Choose an option: 2
Stack elements are: 2   3
```

```
Do you want to continue?(0(no),1(yes)) 0
continue?(0(no),1(yes)) 1
1.Generic stack as integer type         2.Generic stack as string type
Choose an option: 2
Generic stack as string type!!!
1. Push         2. Pop
Choose an option: 1
Enter a string: one
Stack elements are: one
Do you want to continue?(0(no),1(yes)) 1
Generic stack as string type!!!
1. Push         2. Pop
Choose an option: 1
Enter a string: two
Stack elements are: one two
Do you want to continue?(0(no),1(yes)) 1
Generic stack as string type!!!
1. Push         2. Pop
Choose an option: 1
Enter a string: five
Stack elements are: one two     five
Do you want to continue?(0(no),1(yes)) 1
Generic stack as string type!!!
1. Push         2. Pop
Choose an option: 2
Stack elements are: one two
Do you want to continue?(0(no),1(yes)) 0
continue?(0(no),1(yes)) 0
```

RESULT :

Program is successfully executed and output is verified.