Name: Somnath R. Shintre Roll No:

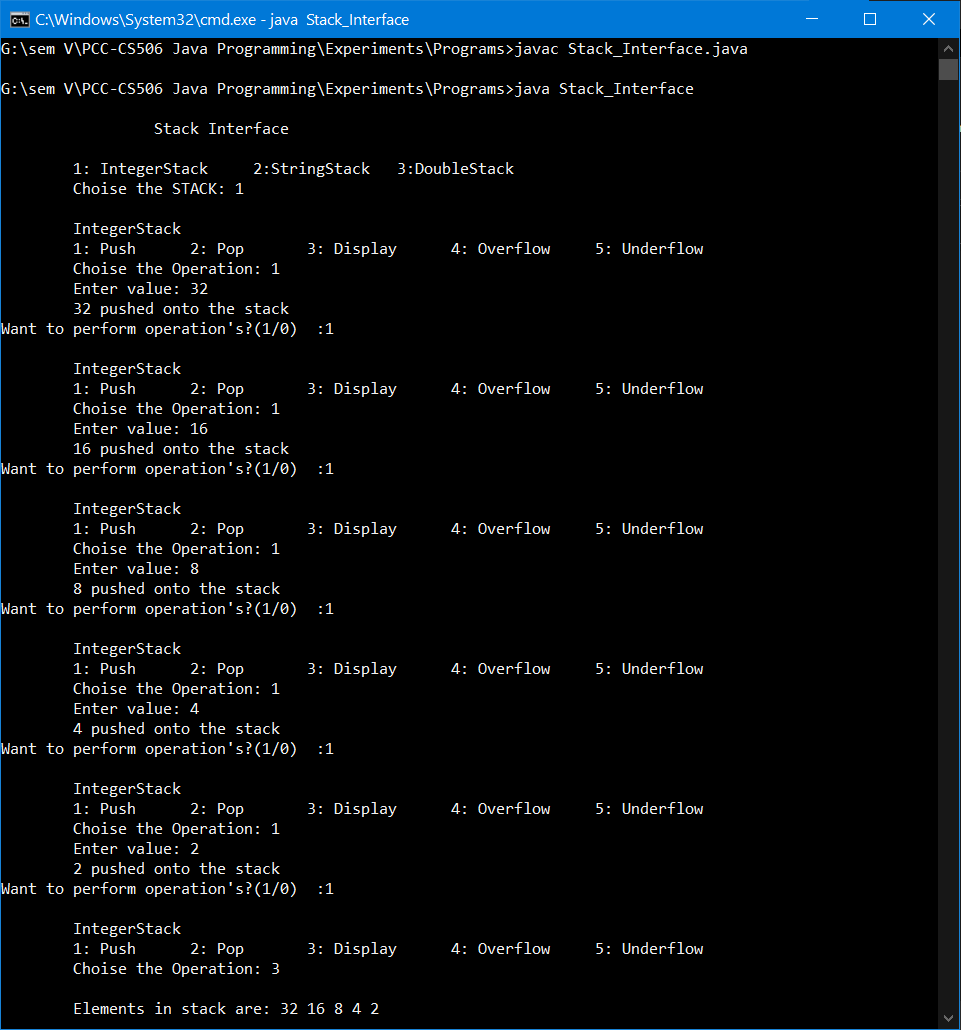
Class: TE CSE Batch:

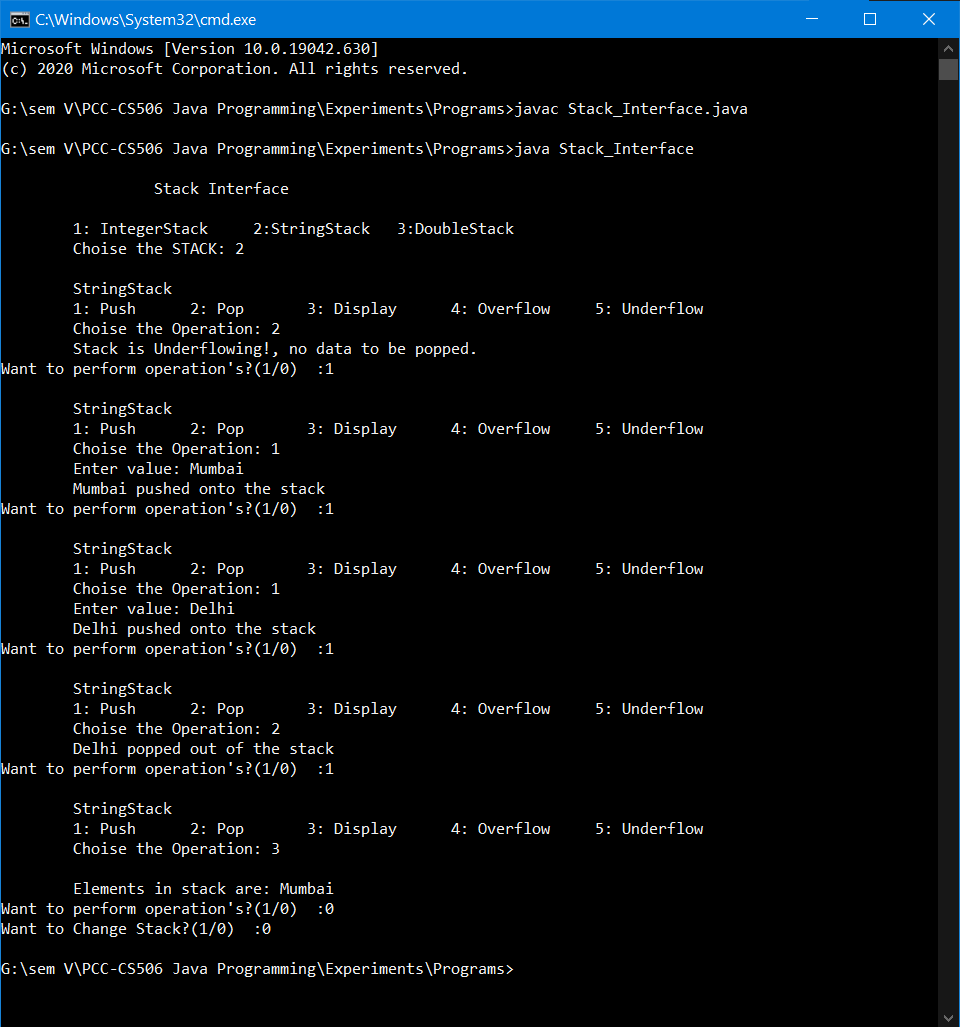
**Title: -** Create the interface stack which has variable size, abstract methods push (), pop (), display (), overflow () and underflow (). We need to implement 3 subclasses IntegerStack, StringStack and DoubleStack respectively by implementing interface. All the methods in interface are declared for string. And in subclass for integerStack convert string to integer. Same thing to all other. Create one test class and check for the working of all the classes.

**Program:-**

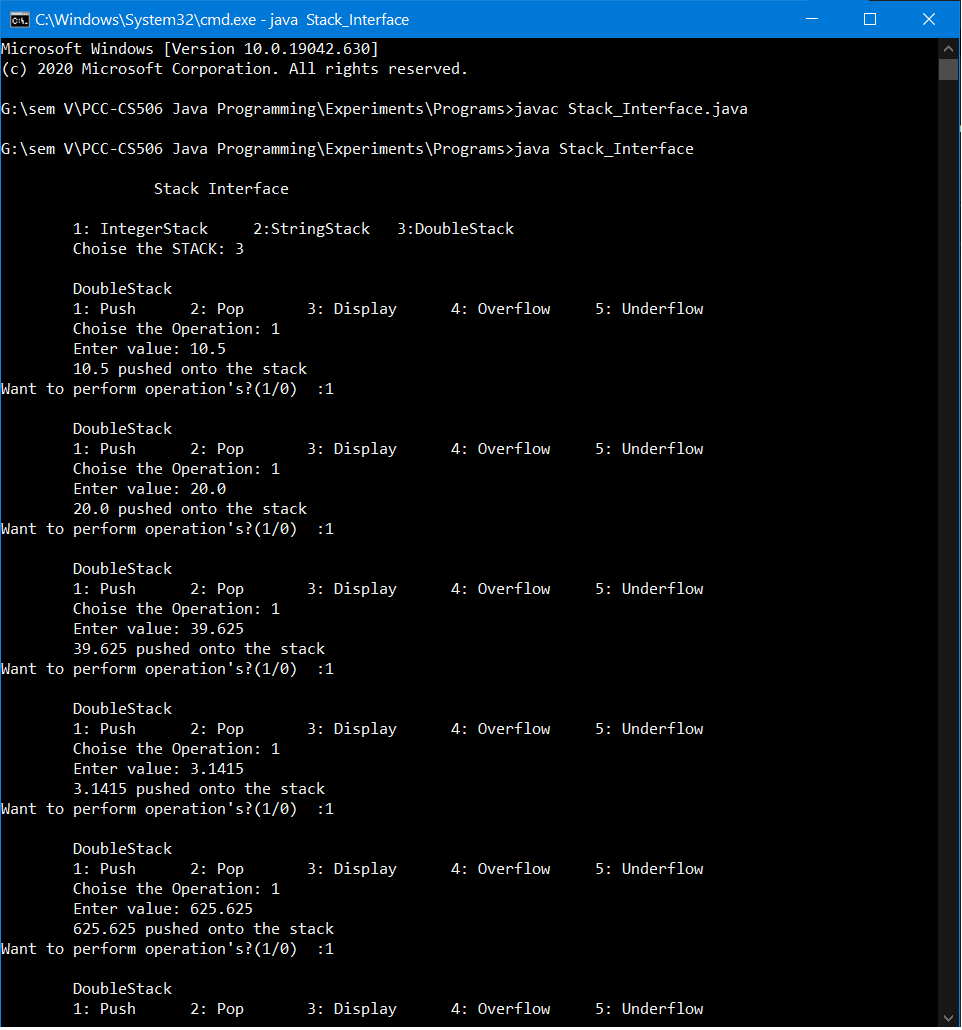
import java.util.Scanner;  
  
/\*\*  
 \* The Stack is an inteface class that holds general data about a stack.  
 \*/  
interface Stack {  
  
 // Instance variable  
 public static final int size = 5;  
  
 // abstract methods  
 // It must be overridden in a subclass.  
 abstract void push(String name);  
  
 abstract void pop();  
  
 abstract void display();  
  
 abstract void overflow();  
  
 abstract void underflow();  
}  
  
class IntegerStack implements Stack {  
  
 // Instance Varibles  
 int arr[] = new int[size];  
 int top = 0;  
  
 @Override  
 public void push(String name) {  
 int n = Integer.parseInt(name);  
 if (top != size) {  
 arr[top] = n;  
 top++;  
 System.out.println("\t"+n + " pushed onto the stack");  
 } else {  
  
 System.out.println("\t"+n + " cannot be pushed, Stack is Overflowing!");  
 }  
 }  
  
 @Override  
 public void pop() {  
 if (top != 0) {  
 int n = arr[top - 1];  
 top--;  
 System.out.println("\t"+n + " popped out of the stack");  
 } else {  
 System.out.println("\tStack is Underflowing!, no data to be popped.");  
 }  
 }  
  
 @Override  
 public void display() {  
 if (top != 0) {  
 System.out.print("\n\tElements in stack are: ");  
 for (int i = 0; i < top; i++) {  
 System.out.print(arr[i] + " ");  
 }  
 System.out.print("\n");  
 } else {  
 System.out.println("\tStack is Underflowing");  
 }  
 }  
  
 @Override  
 public void overflow() {  
 if (top == size) {  
 System.out.println("\tStack is Overflowing!");  
 } else {  
 System.out.println("\tStack is not Overflowing");  
 }  
 }  
  
 @Override  
 public void underflow() {  
 if (top == 0) {  
 System.out.println("\tStack is Underflowing!");  
 } else {  
 System.out.println("\tStack is not Underflowing");  
 }  
 }  
  
}  
  
class StringStack implements Stack {  
  
 // Instance Varibles  
 String arr[] = new String[size];  
 int top = 0;  
  
 @Override  
 public void push(String name) {  
 if (top != size) {  
 arr[top] = name;  
 top++;  
 System.out.println("\t"+name + " pushed onto the stack");  
 } else {  
 System.out.println("\t"+name + "canot be pushed, Stack is Overflowing!");  
 }  
 }  
  
 @Override  
 public void pop() {  
 if (top != 0) {  
 String n = arr[top - 1];  
 top--;  
 System.out.println("\t"+n + " popped out of the stack");  
 } else {  
 System.out.println("\tStack is Underflowing!, no data to be popped.");  
 }  
 }  
  
 @Override  
 public void display() {  
 if (top != 0) {  
 System.out.print("\n\tElements in stack are: ");  
 for (int i = 0; i < top; i++) {  
 System.out.print(arr[i] + " ");  
 }  
 System.out.print("\n");  
 }else {  
 System.out.println("\tStack is Underflowing");  
 }  
 }  
  
 @Override  
 public void overflow() {  
 if (top == size) {  
 System.out.println("\tStack is Overflowing!");  
 } else {  
 System.out.println("\tStack is not Overflowing");  
 }   
 }  
  
 @Override  
 public void underflow() {  
 if (top == 0) {  
 System.out.println("\tStack is Underflowing!");  
 } else {  
 System.out.println("\tStack is not Underflowing");  
 }  
 }  
  
}  
  
class DoubleStack implements Stack {  
  
 // Instance Varibles  
 double arr[] = new double[size];  
 int top = 0;  
  
 @Override  
 public void push(String name) {  
 double n = Double.parseDouble(name);  
 if (top != size) {  
 arr[top] = n;  
 top++;  
 System.out.println("\t"+n + " pushed onto the stack");  
 } else {  
 System.out.println("\t"+n + "canot be pushed, Stack is Overflowing!");  
 }  
 }  
  
 @Override  
 public void pop() {  
 if (top != 0) {  
 double n = arr[top - 1];  
 top--;  
 System.out.println("\t"+n + " popped out of the stack");  
 } else {  
 System.out.println("\tStack is Underflowing!, no data to be popped.");  
 }  
 }  
  
 @Override  
 public void display() {  
 if (top != 0) {  
 System.out.print("\n\tElements in stack are: ");  
 for (int i = 0; i <= top; i++) {  
 System.out.print(arr[i] + " ");  
 }  
 System.out.print("\n");  
 } else {  
 System.out.println("\tStack is Underflowing");  
 }  
 }  
  
 @Override  
 public void overflow() {  
 if (top == size) {  
 System.out.println("\tStack is Overflowing!");  
 } else {  
 System.out.println("\tStack is not Overflowing");  
 }  
 }  
  
 @Override  
 public void underflow() {  
 if (top == 0) {  
 System.out.println("\tStack is Underflowing!");  
 } else {  
 System.out.println("\tStack is not Underflowing");  
 }  
 }  
}  
  
public class Stack\_Interface {  
 public static void main(String[] args) {  
  
 // Creating objects or instances  
 IntegerStack IS = new IntegerStack();  
 StringStack SS = new StringStack();  
 DoubleStack DS = new DoubleStack();  
  
 // Scanner class to get the user's input  
 int stack, operation, con1, con2;  
 Boolean fst = true, snd = true;  
 String data;  
 Scanner sc = new Scanner(System.in);  
  
 // code-  
 System.out.println("\n\t\t Stack Interface");  
 while(snd){  
 System.out.println("\n\t1: IntegerStack 2:StringStack 3:DoubleStack");  
 System.out.print("\tChoise the STACK: ");  
 stack = sc.nextInt();  
 if(stack == 1){   
 while(fst ){  
 System.out.println("\n\tIntegerStack");  
 System.out.println("\t1: Push 2: Pop 3: Display 4: Overflow 5: Underflow");  
 System.out.print("\tChoise the Operation: ");  
 operation = sc.nextInt();  
 if(operation == 1){  
 System.out.print("\tEnter value: ");  
 data = sc.next();  
 IS.push(data);  
 } else if(operation == 2){  
 IS.pop();  
 } else if(operation == 3){  
 IS.display();  
 } else if(operation == 4){  
 IS.overflow();  
 } else if(operation == 5){  
 IS.underflow();  
 }  
 System.out.print("Want to perform operation's?(1/0) :");  
 con2 = sc.nextInt();  
 if(con2 == 1)  
 fst = true ;  
 else {  
 fst = false;  
 }  
 }  
 } else if(stack == 2) {  
 while(fst ){  
 System.out.println("\n\tStringStack");  
 System.out.println("\t1: Push 2: Pop 3: Display 4: Overflow 5: Underflow");;  
 System.out.print("\tChoise the Operation: ");  
 operation = sc.nextInt();  
 if(operation == 1){  
 System.out.print("\tEnter value: ");  
 data = sc.next();  
 SS.push(data);  
 } else if(operation == 2){  
 SS.pop();  
 } else if(operation == 3){  
 SS.display();  
 } else if(operation == 4){  
 SS.overflow();  
 } else if(operation == 5){  
 SS.underflow();  
 }  
 System.out.print("Want to perform operation's?(1/0) :");  
 con2 = sc.nextInt();  
 if(con2 == 1)  
 fst = true ;  
 else {  
 fst = false;  
 }  
 }  
 } else if(stack == 3){  
 while(fst ){  
 System.out.println("\n\tDoubleStack");  
 System.out.println("\t1: Push 2: Pop 3: Display 4: Overflow 5: Underflow");  
 System.out.print("\tChoise the Operation: ");  
 operation = sc.nextInt();  
 if(operation == 1){  
 System.out.print("\tEnter value: ");  
 data = sc.next();  
 DS.push(data);  
 } else if(operation == 2){  
 DS.pop();  
 } else if(operation == 3){  
 DS.display();  
 } else if(operation == 4){  
 DS.overflow();  
 } else if(operation == 5){  
 DS.underflow();  
 }  
 System.out.print("Want to perform operation's?(1/0) :");  
 con2 = sc.nextInt();  
 if(con2 == 1)  
 fst = true ;  
 else {  
 fst = false;  
 }  
 }  
 }  
 System.out.print("Want to Change Stack?(1/0) :");  
 con1 = sc.nextInt();  
 if(con1 == 1)  
 snd = true ;  
 else {  
 snd = false;  
 }  
 }  
 }  
}

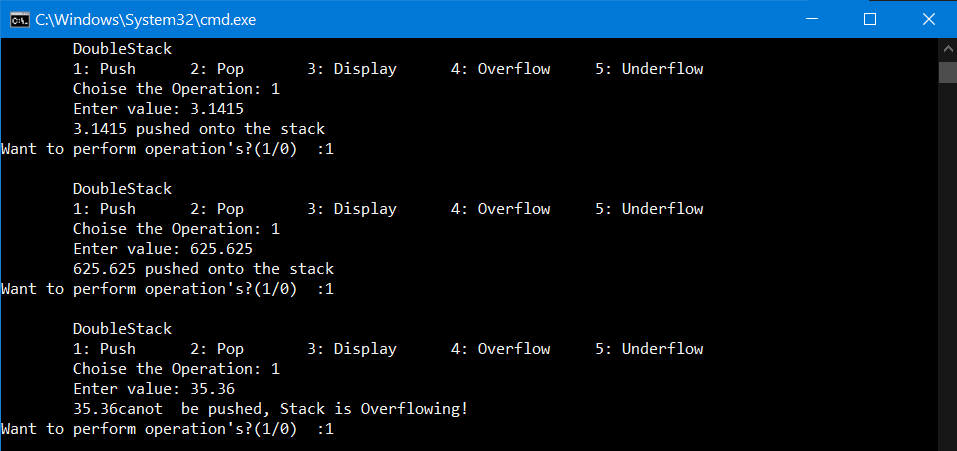
**Output: - 1)**



**2)** 

**3)**

****

****