TERMWORK 7.A

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

**7.2)** Write a JAVA program which has  
  
i. An Interface class for Stack Operations (viz., push(), pop(), peek(),display())  
ii. A Class that implements the Stack Interface and creates a fixed length Stack.  
iii. A Class that implements the Stack Interface and creates a Dynamic Length Stack.  
iv. A Class that uses both the above Stacks through Interface reference and does the Stack operations that demonstrates the runtime binding.

SOURCE CODE:

package addtionaltermwork.termwork7;  
import java.util.Scanner;  
interface stackinterface {

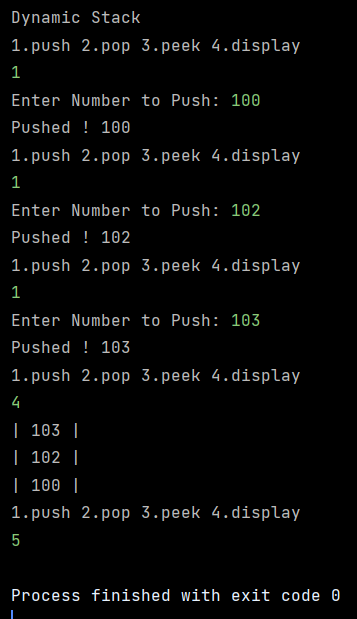
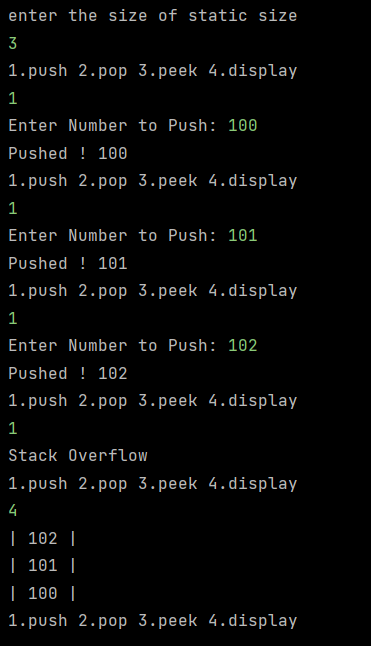
Scanner *read* = new Scanner(System.*in*);  
 void pop();  
 void peek();  
 void push();  
 void display();  
}  
class fixedstack implements stackinterface {  
 int top;  
 int[] stack;

fixedstack(int n) {  
 top = -1;  
 stack = new int[n];  
 }  
  
 public void push() {  
 if (top == stack.length - 1) {  
 System.*out*.println("Stack Overflow");  
 } else {  
 System.*out*.print("Enter Number to Push: ");  
 stack[++top] = *read*.nextInt();  
 System.*out*.println("Pushed ! " + stack[top]);  
 }  
 }  
  
 public void pop() {  
 if (top == -1) {  
 System.*out*.println("Stack Underflow");  
 } else {  
 System.*out*.println("Popped: " + stack[top]);  
 top--;  
 }  
 }  
 public void peek() {  
 System.*out*.println("Element at the top : " + stack[top]);  
 }  
 public void display() {  
 for (int i = top; i >= 0; i--)  
 System.*out*.println("| " + stack[i] + " |");  
 }  
 void operate() {  
 int choice;  
 do {  
 System.*out*.println("1.push 2.pop 3.peek 4.display ");  
 choice = *read*.nextInt();  
 switch (choice) {  
 case 1 -> push();  
 case 2 -> pop();  
 case 3 -> peek();  
 case 4 -> display();  
 }  
 } while (choice != 5);  
 }  
}  
  
class dynamicStack implements stackinterface {  
 int top = -1;  
 int size = 1;  
 int[] stack = new int[1];  
  
 public void increasecapacity() {  
 int[] newStack = new int[size + 1];  
 if (size >= 0) System.*arraycopy*(stack, 0, newStack, 0, size);  
 stack = newStack;  
 size++;  
 }  
 public void push() {  
 increasecapacity();  
 System.*out*.print("Enter Number to Push: ");  
 stack[++top] = *read*.nextInt();  
 System.*out*.println("Pushed ! " + stack[top]);  
 }  
 public void pop() {  
 if (top == -1) {  
 System.*out*.println("Stack Underflow");  
 } else {  
 System.*out*.println("Popped: " + stack[top]);  
 top--;  
 }  
 }  
  
 public void peek() {  
 System.*out*.println("Element at the top : " + stack[top]);  
 }  
  
 public void display() {  
 for (int i = top; i >= 0; i--)  
 System.*out*.println("| " + stack[i] + " |");  
 }  
  
 void operate() {  
 int choice;  
 do {  
 System.*out*.println("1.push 2.pop 3.peek 4.display ");  
 choice = *read*.nextInt();  
 switch (choice) {  
 case 1 -> push();  
 case 2 -> pop();  
 case 3 -> peek();  
 case 4 -> display();  
 }  
 } while (choice != 5);  
 }  
}  
  
public class stackcalss {  
 public static void main(String[] args) {  
 *Scanner read = new Scanner(System.in);  
 int s;  
 System.out.println("enter the size of static size");  
 s = read.nextInt();*

*fixedstack staticstack = new fixedstack(s);  
 staticstack.operate();*

System.*out*.println("Dynamic Stack");  
 dynamicStack dynastack=new dynamicStack();  
 dynastack.operate();  
  
 }  
}

OUTPUT



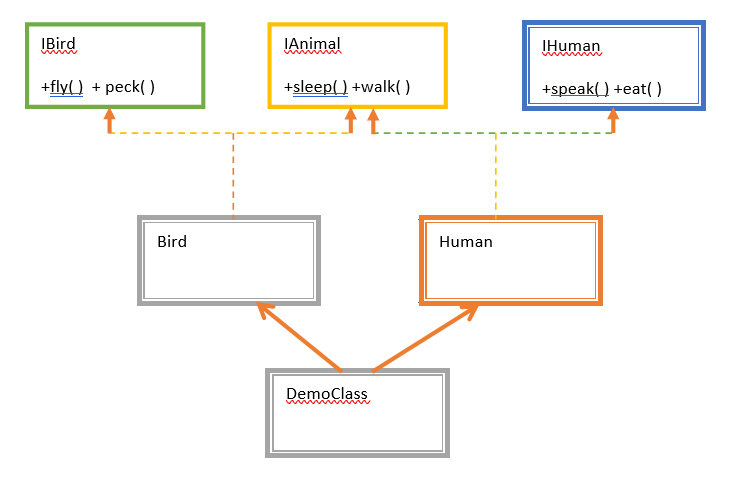
TERMWORK 7.B

NAME-PARISHKAR SINGH

USN – 2GI20CS081

DIV – B

**7.3)** Design an interface IAnimal that has walk, and sleep methods, an interface IBird that has fly, and peck methods, an interface IHuman that has eat and speak methods. Construct a Bird class that implements IAnimal and IBird interfaces and also construct Human class that implements IAnimal and IHuman interfaces. Demonstrate the working of these methods by invoking the methods using appropriate reference variables.



SOURCE CODE:

package practice;  
  
interface IAnimal {  
 void walk();  
  
 void sleep();  
}  
  
interface IBird {  
 void fly();  
  
 void peck();  
}  
  
interface IHuman {  
 void eat();  
  
 void speak();  
}  
  
class Bird implements IAnimal, IBird {  
 public void walk() {  
 System.*out*.println("A bird can walk.");  
 }  
  
 public void sleep() {  
 System.*out*.println("A bird can sleep.");  
 }  
  
 public void fly() {  
 System.*out*.println("A bird can fly.");  
 }  
  
 public void peck() {  
 System.*out*.println("A bird can peck.");  
 }  
}  
  
class Human implements IAnimal, IHuman {  
 public void walk() {  
 System.*out*.println("A human being can walk.");  
 }  
  
 public void sleep() {  
 System.*out*.println("A human being can sleep.");  
 }  
  
 public void eat() {  
 System.*out*.println("A human being can eat.");  
 }  
  
 public void speak() {  
 System.*out*.println("A human being can speak.");  
 }  
}  
  
public class Seven {  
 public static void main(String[] args) {  
 Bird b = new Bird();  
 Human h = new Human();  
 b.walk();  
 b.sleep();  
 b.fly();  
 b.peck();  
 System.*out*.println("");  
 h.walk();  
 h.sleep();  
 h.eat();  
 h.speak();  
 }  
}

