**Name:** Shradha Mallikarjun Patil

**USN:** 2GI20CS144

**TERMWORK 7.2**

**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.

**Program:**

interface Stack

{

public void push(int ele);

public void pop();

public void peek();

public void disp();

}

class FixedStack implements Stack

{

int s[],top;

FixedStack(int size)

{

s=new int[size];

top=-1;

}

public void push(int ele)

{

if(top==s.length-1)

{

System.out.println("Stack overflow.");

}

else{

s[++top]=ele;

}

}

public void pop()

{

int d;

if(top==-1)

System.out.println("Stack empty.");

else

{

d=s[top--];

System.out.println(d+" is popped.");

}

}

public void peek()

{

if(top==-1)

System.out.println("Stack empty.");

else

{

System.out.println(s[top]+" is top element of stack.");

}

}

public void disp()

{

if(top==-1)

{

System.out.println("Empty stack");

}

else

{

for(int i=top;i>=0;i--)

{

System.out.println(s[i]);

}

}

}

}

class DynamicStack implements Stack

{

int top, s[];

int size;

DynamicStack(int size)

{

this.size=size;

s=new int[size];

top=-1;

}

public void resize(int nsize)

{

int tempstk[]=new int[nsize];

for(int i=0;i<s.length;i++)

{

tempstk[i]=s[i];

s=tempstk;

}

size=nsize;

}

public void push(int ele)

{

if(top==s.length-1)

{

System.out.println("Re-sizing...");

resize(size\*2);

push(ele);

}

else

{

s[++top]=ele;

}

}

public void pop()

{

if(top==-1)

{

System.out.println("Empty stack");

}

else

{

System.out.println(s[top--]+" is popped");

}

}

public void peek()

{

if(top==-1)

{

System.out.println("Empty stack");

}

else

{

System.out.println(s[top]+" is top element of stack");

}

}

public void disp()

{

if(top==-1)

{

System.out.println("Empty stack");

}

else

{

for(int i=top;i>=0;i--)

{

System.out.println(s[i]);

}

}

}

}

class Main

{

public static void main(String args[])

{

FixedStack f=new FixedStack(5);

f.push(5);

f.push(8);

f.push(7);

f.push(6);

f.push(2);

f.push(3);

System.out.println("Stack elements are:");

f.disp();

f.peek();

f.pop();

f.pop();

f.pop();

System.out.println("Stack elements are:");

f.disp();

System.out.println("");

DynamicStack d=new DynamicStack(3);

d.push(5);

d.push(8);

d.push(7);

System.out.println("Stack elements are:");

d.disp();

d.peek();

d.pop();

d.pop();

d.pop();

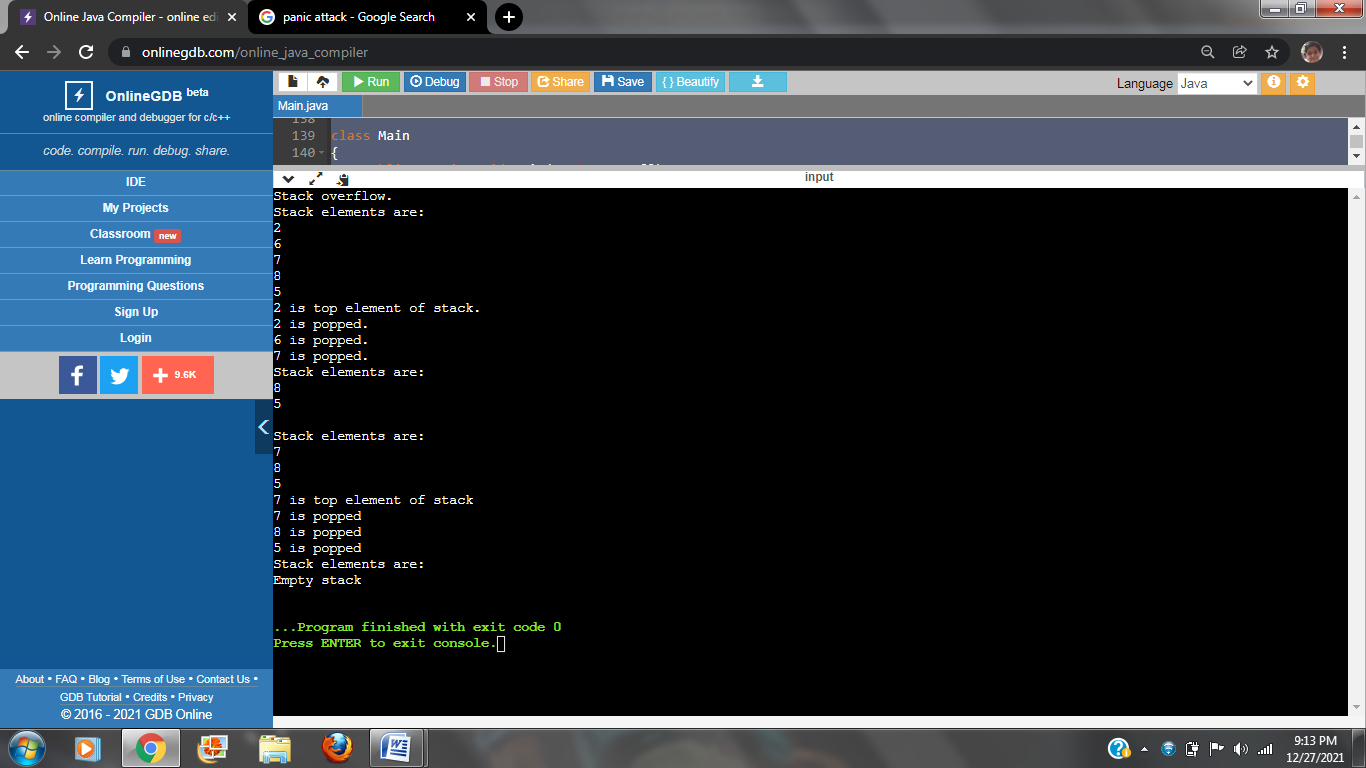
System.out.println("Stack elements are:");

d.disp();

}

}

**OUTPUT:**

****

**TERMWORK 7.3:**

**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.

DemoClass

Human

Bird

IBird

+fly( ) + peck( )

IAnimal

+sleep( ) +walk( )

IHuman

+speak( ) +eat( )

**CODE:**

**interface IAnimal**

**{**

**public void sleep();**

**public void walk();**

**}**

**interface IBird**

**{**

**public void fly();**

**public void peck();**

**}**

**interface IHuman**

**{**

**public void speak();**

**public void eat();**

**}**

**class Bird implements IAnimal,IBird**

**{**

**public void sleep()**

**{**

**System.out.println("Birds sleep but not in their nests.");**

**}**

**public void walk()**

**{**

**System.out.println("Birds generally hop.");**

**}**

**public void fly()**

**{**

**System.out.println("Most birds can fly.");**

**}**

**public void peck()**

**{**

**System.out.println("Birds peck and can potentially kill you.");**

**}**

**}**

**class Human implements IAnimal,IHuman**

**{**

**public void sleep()**

**{**

**System.out.println("Most humans can sleep, while insomniacs can't.");**

**}**

**public void walk()**

**{**

**System.out.println("Humans walk.");**

**}**

**public void speak()**

**{**

**System.out.println("Humans can comprehend and talk various languages.");**

**}**

**public void eat()**

**{**

**System.out.println("Humans are omnivorous, cruel beasts.");**

**}**

**}**

**class Main**

**{**

**public static void main(String args[])**

**{**

**Bird b=new Bird();**

**b.sleep();**

**b.walk();**

**b.fly();**

**b.peck();**

**System.out.println("");**

**Human h=new Human();**

**h.sleep();**

**h.walk();**

**h.speak();**

**h.eat();**

**}**

**}**

**OUTPUT:**

