

|||||||||||||||||||||||||||||||||||||||||||||

Q1

```
class Demo{
    public static void main(String args[]){
        int[] ar=new int[3];
        //
        //
        Customer c1=new Customer();
    }
}
```

|||||||||||||||||||||||||||||||||||||||||

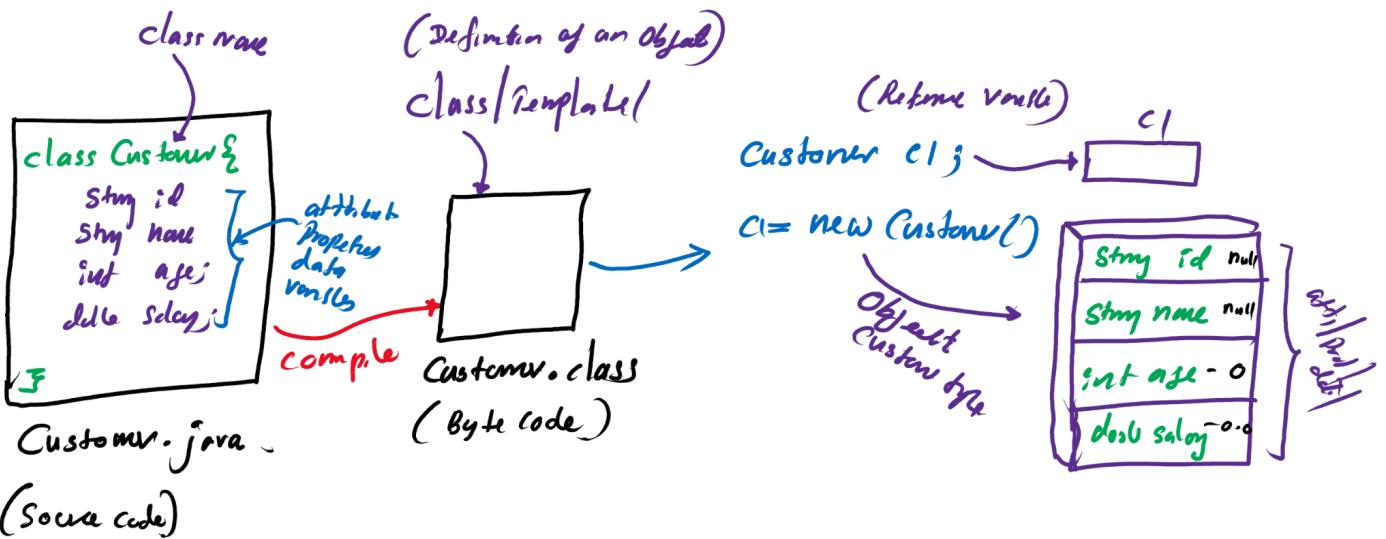
Q2

//-----Customer.java-----

```
class Customer{
    String id;
    String name;
    int age;
    double salary;
}
```

//-----Demo.java-----

```
class Demo{
    public static void main(String args[]){
        int[] ar=new int[3];
        //
        //
        Customer c1=new Customer();
    }
}
```



Q3 Accessing properties

```

class Demo{
    public static void main(String args[]){
        int[] ar=new int[3];
        ar[0]=100;
        ar[1]=200;

        Customer c1=new Customer();
        c1.id="C001";
        c1.name="Niroth";
        c1.age=10;
        c1.salary=3400;

        System.out.println("id : "+c1.id);
        System.out.println("name : "+c1.name);
        System.out.println("age : "+c1.age);
        System.out.println("salary : "+c1.salary);
    }
}

```

```
///////////////////////////////  
Q4  
class Customer{  
    String id;  
    String name;  
    int age;  
    double salary;  
}  
class Demo{  
    public static void main(String args[]){  
        int[] ar=new int[3];  
        ar[0]=100;  
        ar[1]=200;  
  
        Customer c1=new Customer();  
        c1.id="C001";  
        c1.name="Niroth";  
        c1.age=10;  
        c1.salary=3400;  
  
        System.out.println("id : "+c1.id);  
        System.out.println("name : "+c1.name);  
        System.out.println("age : "+c1.age);  
        System.out.println("salary : "+c1.salary);  
    }  
}
```

```
///////////////////////////////  
Q5 Adding java methods  
=====  
class Customer{  
    private String id;  
    private String name;  
    private int age;  
    private double salary;  
}  
class Demo{  
    public static void main(String args[]){  
        Customer c1=new Customer();  
        c1.id="C001";  
        c1.name="Niroth";  
        c1.age=10;  
        c1.salary=3400;  
  
        System.out.println("id : "+c1.id);  
        System.out.println("name : "+c1.name);  
        System.out.println("age : "+c1.age);  
        System.out.println("salary : "+c1.salary);  
    }  
}
```

Q6 From Q5 (Setters and getters)

Q6 From Q5 (Setters and getters)

```
class Customer{  
    private String id;  
    private String name;  
    private int age;  
    private double salary;  
  
    public void setId(String id){  
        this.id=id;  
    }  
    public void setName(String name){  
        this.name=name;  
    }  
    public void setAge(int age){  
        this.age=age;  
    }  
    public void setSalary(double salary){  
        this.salary=salary;  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Customer c1=new Customer();  
        c1.setId("C001");           //c1.id="C001";  
        c1.setName("Nirot");       //c1.name="Niroth";  
        c1.setAge(10);             //c1.age=10;  
        c1.setSalary(3400);        //c1.salary=3400;  
  
        System.out.println("id   : "+c1.id);  
        System.out.println("name  : "+c1.name);  
        System.out.println("age   : "+c1.age);  
        System.out.println("salary : "+c1.salary);  
    }  
}
```

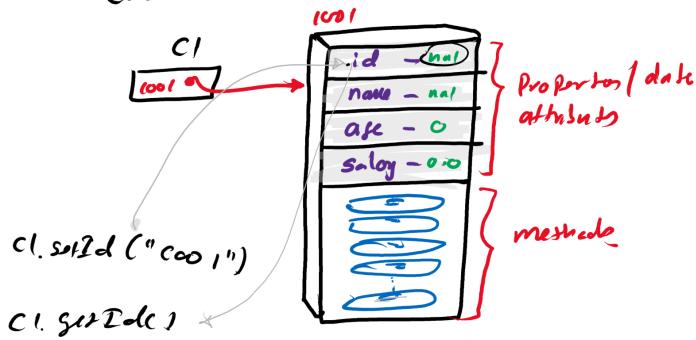
```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Q7 From Q6

=====

```
class Customer{  
    private String id;  
    private String name;  
    private int age;  
    private double salary;  
  
    public void setId(String id){  
        this.id=id;  
    }  
    public void setName(String name){  
        this.name=name;  
    }  
    public void setAge(int age){  
        this.age=age;  
    }  
    public void setSalary(double salary){  
        this.salary=salary;  
    }  
    public String getId(){  
        return id;  
    }  
    public String getName(){  
        return name;  
    }  
    public int getAge(){  
        return age;  
    }  
    public double getSalary(){  
        return salary;  
    }  
}  
  
class Demo{  
    public static void main(String args[]){  
        Customer c1=new Customer();  
        c1.setId("C001");      //c1.id="C001";  
        c1.setName("Nirot");//c1.name="Niroth";  
        c1.setAge(10);         //c1.age=10;  
        c1.setSalary(3400);    //c1.salary=3400;  
  
        System.out.println("id : "+c1.getId());  
        System.out.println("name : "+c1.getName());  
        System.out.println("age : "+c1.getAge());  
        System.out.println("salary : "+c1.getSalary());  
    }  
}
```

Customer c1 = new Customer()



//////////

Q8 Exercise

=====

```
class Demo{
```

```
    public static void main(String args[]){
        int length;
        int width;
        int height;

        length=12;
        width=5;
        height=3;

        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}
```

//////////

Q9 From Q8

=====

```
class Demo{
```

```
    public static void main(String args[]){
        /*int length;
        int width;
        int height;*/
        Box b1=new Box();

        length=12;
        width=5;
        height=3;

        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume); //prints "Volume : 180"
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q10 From Q9
=====
class Box{
    int length; //Attribute declaration
    int width;    //Attribute declaration
    int height;   //Attribute declaration
}
class Demo{
    public static void main(String args[]){
        /*int length;
        int width;
        int height;*/
        Box b1=new Box();

        b1.length=12;
        b1.width=5;
        b1.height=3;

        int volume;
        volume=b1.length*b1.width*b1.height;
        System.out.println("Volume : "+volume); //prints "Volume : 180"
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q11 Adding methods (From Q10)
=====
class Box{
    int length;    //Attribute declaration
    int width;    //Attribute declaration
    int height;   //Attribute declaration
}
class Demo{
    public static void main(String args[]){
        Box b1=new Box();

        /*b1.length=12;
        b1.width=5;
        b1.height=3;*/
        b1.setSize(12,5,3);

        int volume;
        volume=b1.length*b1.width*b1.height;
        System.out.println("Volume : "+volume); //prints "Volume : 180"
    }
}
```

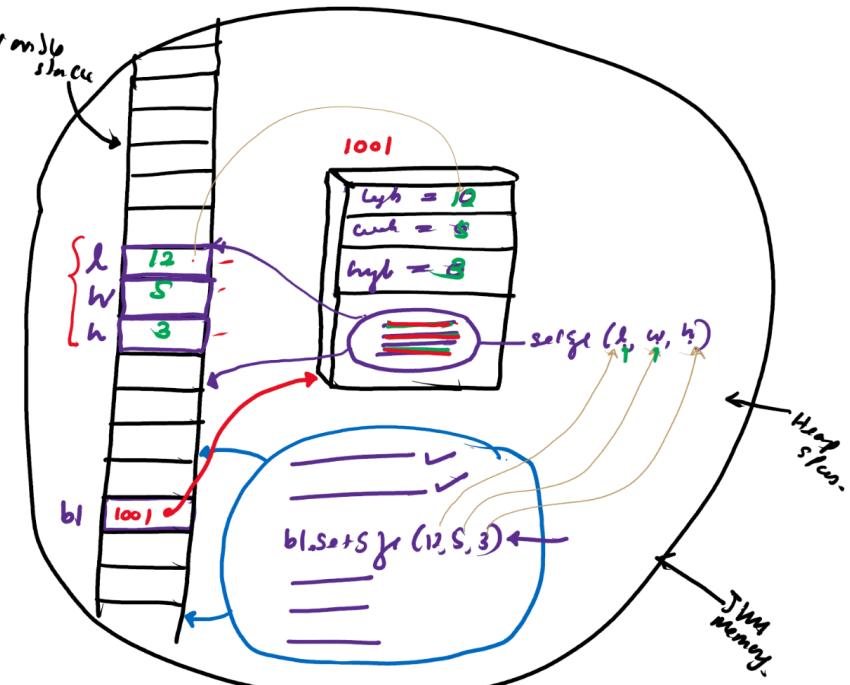
```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Q12 From Q11

=====

```
class Box{  
    int length; //Attribute declaration  
    int width; //Attribute declaration  
    int height; //Attribute declaration  
  
    //-----Methods declaration-----  
    public void setSize(int l, int w, int h){  
        length=l;  
        width=w;  
        height=h;  
    }  
}  
  
class Demo{  
    public static void main(String args[]){  
        Box b1=new Box();  
  
        b1.setSize(12,5,3);  
  
        int volume;  
        volume=b1.length*b1.width*b1.height;  
        System.out.println("Volume : "+volume); //prints "Volume : 180"  
    }  
}
```

```
class Box{  
    int length;  
    int width;  
    int height;  
  
    public void setSize(int l, int w, int h){  
        length=l; obj l=12  
        width=w; w=5  
        height=h; h=3  
    }  
}  
  
class Demo{  
    public static void main(String args[]){  
        Box b1; b1  
        b1=new Box(); b1  
        b1.setSize(12,5,3);  
        int volume;  
        volume=b1.length*b1.width*b1.height;  
        System.out.println("Volume : "+volume);  
    }  
}
```



//////////

Q13 Exercise

=====

```
class Box{
    int length;
    int width;
    int height;

    public void setSize(int l, int w, int h){
        length=l;
        width=w;
        height=h;
    }
}

class Demo{
    public static void main(String args[]){
        Box b1;
        b1=new Box();
        b1.setSize(12,5,3);

        /*int volume;
        volume=b1.length*b1.width*b1.height;
        System.out.println("Volume : "+volume);*/
        b1.printVolume();
    }
}
```

//////////

Q14 From Q13

=====

```
class Box{
    int length;
    int width;
    int height;

    public void setSize(int l, int w, int h){
        length=l;
        width=w;
        height=h;
    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}
```

```

class Demo{
    public static void main(String args[]){
        Box b1;
        b1=new Box();
        b1.setSize(12,5,3);
        b1.printVolume();
    }
}

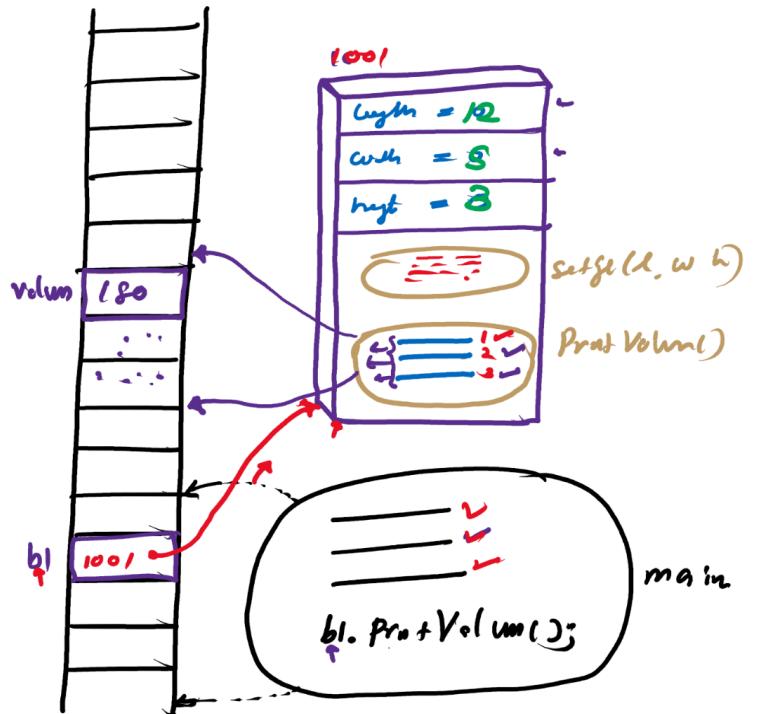
class Box{
    int length;
    int width;
    int height;

    public void setSize(int l, int w, int h){
        length=l;
        width=w;
        height=h;
    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}

class Demo{
    public static void main(String args[]){
        Box b1; ✓
        b1=new Box(); ✓
        b1.setSize(12,5,3); ✓
        b1.printVolume();
    }
}

```



Q15 Exercise

```

class Box{
    int length;
    int width;
    int height;

    public void setSize(int l, int w, int h){
        length=l;
        width=w;
        height=h;
    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}

```

```
class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        b1.setSize(12,5,3);
        b1.printVolume();
        b1.printArea(); //Area : 222
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q16 From Q15

=====

```
class Box{
    int length;
    int width;
    int height;

    public void setSize(int l, int w, int h){
        length=l;
        width=w;
        height=h;
    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }

    public void printArea(){
        int area;
        area=2*length*width+2*length*height+2*width*height;
        System.out.println("Area : "+area);
    }
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        b1.setSize(12,5,3);
        b1.printVolume();
        b1.printArea(); //Area : 222
    }
}
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Q17 Exercise

```
=====
```

```
class Box{  
    int length;  
    int width;  
    int height;  
    public void setSize(int length, int width, int height){  
        length=length;  
        width=width;  
        height=height;  
    }  
    public void printVolume(){  
        int volume;  
        volume=length*width*height;  
        System.out.println("Volume : "+volume);  
    }  
    public void printArea(){  
        int area;  
        area=2*length*width+2*length*height+2*width*height;  
        System.out.println("Area : "+area);  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Box b1=new Box();  
        b1.setSize(12,5,3);  
        b1.printVolume();  
        System.out.println("length : "+b1.length);  
        System.out.println("width : "+b1.width);  
        System.out.println("height : "+b1.height);  
    }  
}
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Q18

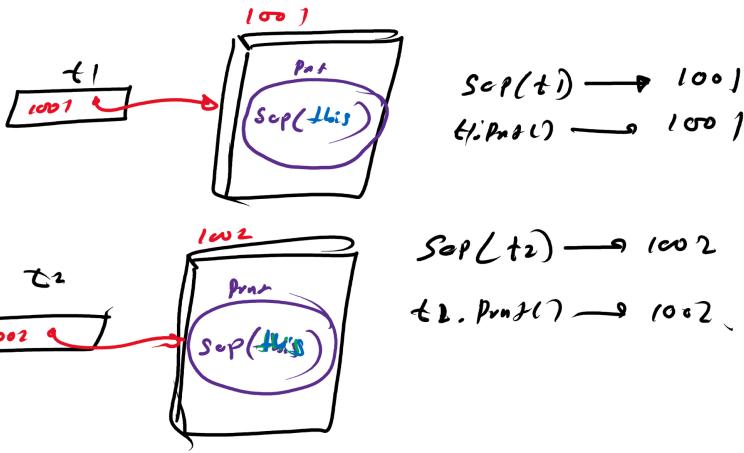
```
class Test{
```

```
    public void print(){  
        System.out.println(this);  
    }  
}
```

```
}
```

```
class Demo{
```

```
    public static void main(String args[]){  
        Test t1=new Test();  
        System.out.println(t1); //  
        t1.print();  
        System.out.println();  
        Test t2=new Test();  
        System.out.println(t2); //  
        t2.print();  
    }  
}
```



//////////////////////////////

Q19 From Q17 Using "this"

=====

```

class Box{
    int length;
    int width;
    int height;
    public void setSize(int length, int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
    public void printArea(){
        int area;
        area=2*length*width+2*length*height+2*width*height;
        System.out.println("Area : "+area);
    }
}
class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        b1.setSize(12,5,3);
        b1.printVolume();
        System.out.println("length : "+b1.length);
        System.out.println("width : "+b1.width);
        System.out.println("height : "+b1.height);
    }
}

```

Object Initialization (Constructors)

Q20 Default Constructor

```
=====
class Box{
    int length;
    int width;
    int height;
    /*compiler inserts
    Box(){
        length=0;
        width=0;
        height=0;
    }*/
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        System.out.println("length : "+b1.length);
        System.out.println("width : "+b1.width);
        System.out.println("height : "+b1.height);
    }
}
```

Q21

```
=====
class Box{
    int length;
    int width;
    int height;
    Box(){
        length=1;
        width=1;
        height=1;
    }
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        System.out.println("length : "+b1.length);
        System.out.println("width : "+b1.width);
        System.out.println("height : "+b1.height);
    }
}
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Q22 Parameterized Constructors

```
=====
```

```
class Box{  
    int length;  
    int width;  
    int height;  
  
    Box(int length,int width, int height){  
        this.length=length;  
        this.width=width;  
        this.height=height;  
    }  
    public void printVolume(){  
        int volume;  
        volume=length*width*height;  
        System.out.println("Volume : "+volume);  
    }  
}  
  
class Demo{  
    public static void main(String args[]){  
        Box b1=new Box(12,5,3);  
        System.out.println("length : "+b1.length);  
        System.out.println("width : "+b1.width);  
        System.out.println("height : "+b1.height);  
        b1.printVolume();  
    }  
}
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Q23 Constructor Overloading

```
=====
```

```
class Box{  
    int length;  
    int width;  
    int height;  
  
    Box(int length,int width, int height){  
        this.length=length;  
        this.width=width;  
        this.height=height;  
        System.out.println("Box(int,int,int)");  
    }  
    Box(int length){  
        this.length=length;  
        this.width=length;  
        this.height=length;  
        System.out.println("Box(int)");  
    }  
}
```

```

Box(){
    this.length=1;
    this.width=1;
    this.height=1;
    System.out.println("Box()");
}
public void printVolume(){
    int volume;
    volume=length*width*height;
    System.out.println("Volume : "+volume);
}

}
class Demo{
    public static void main(String args[]){
        Box b1=new Box(12,5,3);
        System.out.println("length : "+b1.length);
        System.out.println("width : "+b1.width);
        System.out.println("height : "+b1.height);
        b1.printVolume();
        System.out.println();

        Box b2=new Box(10);
        System.out.println("length : "+b2.length);
        System.out.println("width : "+b2.width);
        System.out.println("height : "+b2.height);
        b2.printVolume();
        System.out.println();

        Box b3=new Box();
        System.out.println("length : "+b3.length);
        System.out.println("width : "+b3.width);
        System.out.println("height : "+b3.height);
        b3.printVolume();
    }
}

```

//////////

Q24 Exercise

=====

```

class Box{
    int length;
    int width;
    int height;

    Box(int length,int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
        System.out.println("Box(int,int,int)");
    }
}
```

```

public void printVolume(){
    int volume;
    volume=length*width*height;
    System.out.println("Volume : "+volume);
}

}

class Demo{
    public static void main(String args[]){
        Box b1=new Box(12,5,3);
        System.out.println("length : "+b1.length);
        System.out.println("width : "+b1.width);
        System.out.println("height : "+b1.height);
        b1.printVolume();
        System.out.println();

        Box b3=new Box(); //Illegal ?
    }
}

```

//////////
Q25 Exercise

```

=====
class Box{
    int length;
    int width;
    int height;

    Box(){
        //compiler inserts
    }
    Box(int length,int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
        System.out.println("Box(int,int,int)");
    }
    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        b1.Box(12,5,3); //Illegal
    }
}
```

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Q26 Exercise

=====

```
class Box{  
    int length;  
    int width;  
    int height;  
  
    public Box(){  
        //compiler inserts  
    }  
    public void Box(int length,int width, int height){  
        this.length=length;  
        this.width=width;  
        this.height=height;  
        System.out.println("Box(int,int,int)");  
    }  
    public void printVolume(){  
        int volume;  
        volume=length*width*height;  
        System.out.println("Volume : "+volume);  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Box b1=new Box();  
        b1.Box(12,5,3); //Legal  
        b1.printVolume();  
  
        Box b2=new Box(12,5,3); //Illegal  
    }  
}
```

```
////////////////////////////////////////////////////////////////////////
Q27
class Box{
    int length;
    int width;
    int height;

    Box(int length,int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
        System.out.println("Box(int,int,int)");
    }
    Box(int length){
        /*this.length=length;
        this.width=length;
        this.height=length;*/
        //Box(length,length,length);
        this(length,length,length);
        System.out.println("Box(int)");
    }
    Box(){
        /*this.length=1;
        this.width=1;
        this.height=1;*/
        this(1);
        System.out.println("Box()");
    }
    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}
class Demo{
    public static void main(String args[]){
        Box b1=new Box();
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q28 Declaration Values

=====

```
class Box{
    int length=12;
    int width=5;
    int height=3;
    /*Compiler inserts
    Box(){
        length=12;
        width=5;
        height=3;
    }
    */
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        System.out.println("length : "+b1.length);
        System.out.println("width : "+b1.width);
        System.out.println("height : "+b1.height);
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q29

```
class Box{
    int length=12;
    int width=5;
    int height=3;
    Box(){
        /*Compiler inserts
        length=12;
        width=5;
        height=3;
        */
        length=1;
        width=2;
        height=3;
    }
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        System.out.println("length : "+b1.length); //1
        System.out.println("width : "+b1.width); //2
        System.out.println("height : "+b1.height); //3
    }
}
```

//////////

Q30

```
class Box{
    int length=12;
    int width=5;
    int height=3;
    Box(){
        length=1;
        width=2;
        height=3;
    }
    Box(int length,int width, int height){

    }
}
```

class Demo{

```
    public static void main(String args[]){
        Box b1=new Box();
        System.out.println("length : "+b1.length); //1
        System.out.println("width : "+b1.width); //2
        System.out.println("height : "+b1.height); //3
        System.out.println();

        Box b2=new Box(0,0,0);
        System.out.println("length : "+b2.length); //
        System.out.println("width : "+b2.width); //
        System.out.println("height : "+b2.height); //
    }
}
```

//////////

Q31 Exercise

=====

```
class Box{
    int length;
    int width;
    int height;

    length=1;
    width=2;
    height=3;

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

}
```

```
////////////////////////////////////////////////////////////////////////
Q32 Instance Blocks
=====
class Box{
    int length;
    int width;
    int height;
    {
        length=1;
        width=2;
        height=3;
        System.out.println("instance block");
    }
    Box(){
        //
    }
}
class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        //
        //
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q33 Exercise
=====
class Demo{
    public static void main(String args[]){
        Cylinder c1=new Cylinder();
        c1.setLength(12.5);
        c1.setRadius(5.0);
        c1.printVolume();
        c1.printArea();

        Cylinder c2=new Cylinder();
        c2.setDimensions(5.0,1.5);
        c2.printVolume();
        c2.printArea();

        Cylinder c3=new Cylinder(1.2,.5);
        System.out.println("Volume of c3 : "+c3.getVolume());
        System.out.println("Area of c3 : "+c3.getArea());
        System.out.println("Length of c3 : "+c3.getLength());
        System.out.println("Radius of c3 : "+c3.getRadius());

    }
}
```



```

Cylinder c2=new Cylinder();
c2.setDimensions(5.0,1.5);
c2.printVolume();
c2.printArea();

Cylinder c3=new Cylinder(1.2,.5);
System.out.println("Volume of c3 : "+c3.getVolume());
System.out.println("Area of c3 : "+c3.getArea());
System.out.println("Length of c3 : "+c3.getLength());
System.out.println("Radius of c3 : "+c3.getRadius());

}

}

```

//

Q35 Passing an Object to a method

=====

```

class Box{
    int length;
    int width;
    int height;

    Box(int length,int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    Box(){

    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box(12,5,3);
        b1.printVolume(); //180

        Box b2=new Box();
        b2.printVolume(); //0

        b2.copyValuesFrom(b1); //b2.setSize(b1);
        b2.printVolume(); //180

    }
}

```

```
////////////////////////////////////////////////////////////////////////
Q36 From Q35
=====
class Box{
    int length;
    int width;
    int height;

    Box(int length,int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    Box(){
    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
    public void copyValuesForm(Box x){
        this.length=x.length;
        this.width=x.width;
        this.height=x.height;
    }
}
class Demo{
    public static void main(String args[]){
        Box b1=new Box(12,5,3);
        b1.printVolume(); //180

        Box b2=new Box();
        b2.printVolume(); //0

        b2.copyValuesForm(b1); //b2.setSize(b1);
        b2.printVolume(); //180
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q37 Exercise
=====
class Account{
    String name;
    double balance;
    Account(String name, double balance){
        this.name=name;
        this.balance=balance;
    }
    public void printBalance(){
        System.out.println("Balance : "+balance);
    }
}
class AccountController{
    public void withdraw(double balance, double amount){
        balance-=amount;
    }
}
class Demo{
    public static void main(String args[]){
        Account c1=new Account("Amal",500000);
        c1.printBalance();

        AccountController controller=new AccountController();
        controller.withdraw(c1.balance,10000);

        c1.printBalance();
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q38 From Q37
=====
class Account{
    String name;
    double balance;
    Account(String name, double balance){
        this.name=name;
        this.balance=balance;
    }
    public void printBalance(){
        System.out.println("Balance : "+balance);
    }
}
class AccountController{
    public void withdraw(Account ac, double amount){
        ac.balance-=amount;
    }
}
```

```

class Demo{
    public static void main(String args[]){
        Account c1=new Account("Amal",500000);
        c1.printBalance();

        AccountController controller=new AccountController();
        controller.withdraw(c1,10000);

        c1.printBalance();
    }
}

```

//

Q39 Returning an Object

=====

```

class Box{
    int length;
    int width;
    int height;

    Box(int length,int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    Box(){

    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box(12,5,3);
        b1.printVolume(); //180

        Box b2=b1.getCopy();
        b2.printVolume(); //180

        System.out.println(b1==b2); //false
    }
}

```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q40 From Q39

=====

```
class Box{
    int length;
    int width;
    int height;

    Box(int length,int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    Box(){
    }

    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
    public Box getCopy(){
        Box b1=new Box(length,width,height);
        return b1;
    }
}
class Demo{
    public static void main(String args[]){
        Box b1=new Box(12,5,3);
        b1.printVolume(); //180

        Box b2=b1.getCopy();
        b2.printVolume(); //180

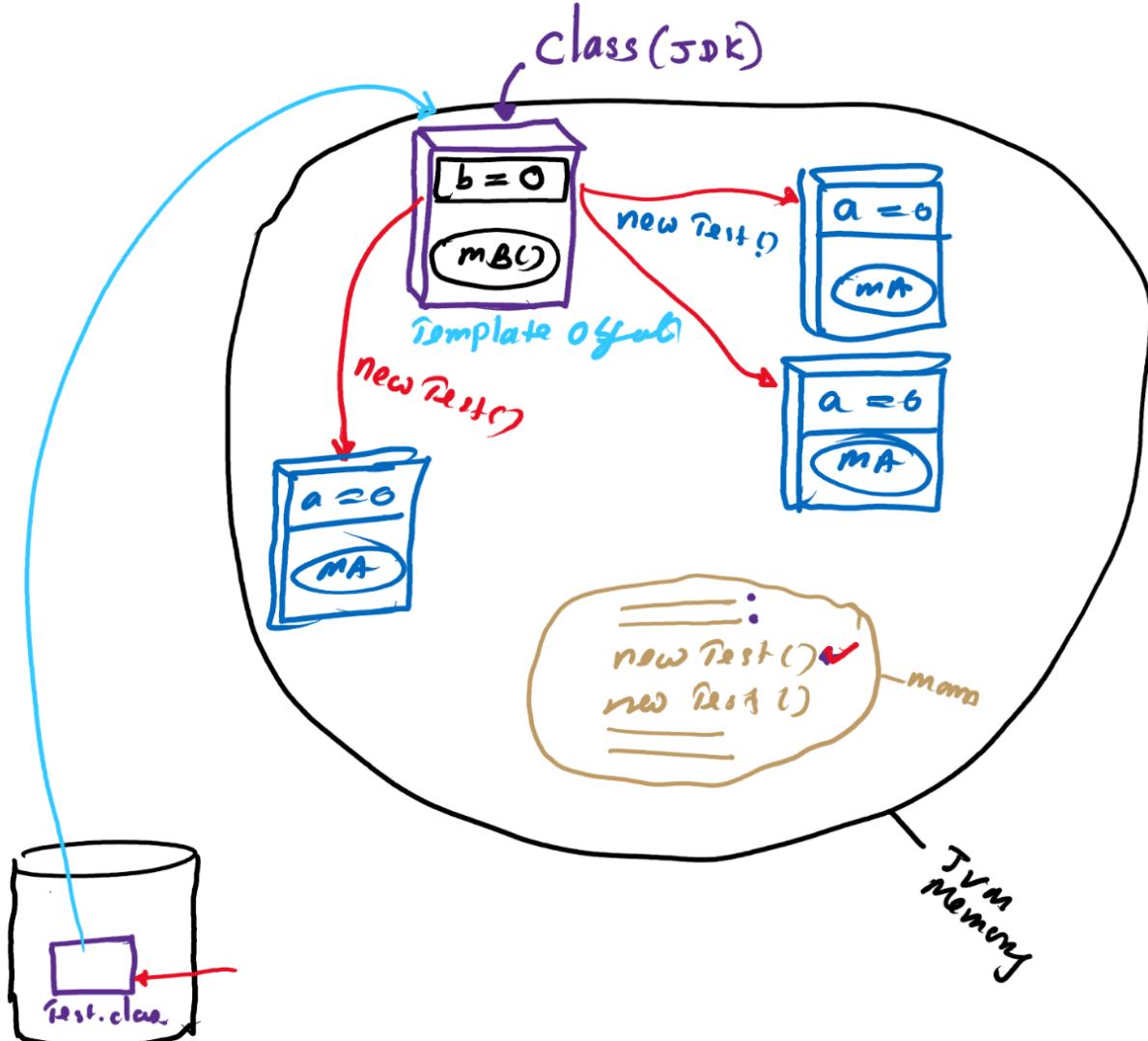
        System.out.println(b1==b2); //false
    }
}
```

Static Initialization

Q41

```
class Test{  
    int a;  
    static int b;  
  
    public void mA(){  
        System.out.println("instance method");  
    }  
    public static void mB(){  
        System.out.println("static method");  
    }  
}  
  
class Demo{  
    public static void main(String args[]){  
        Test t1=new Test();  
        Test t2=new Test();  
    }  
}
```

;



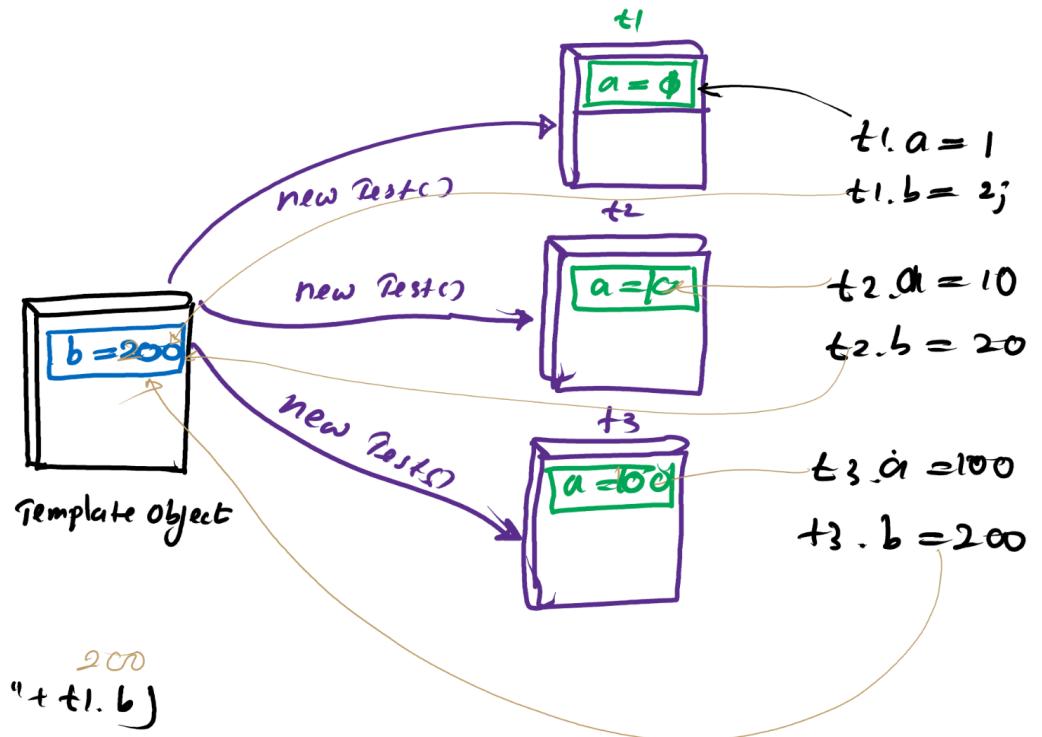
```
Q42
class Test{
    int a; //Instance variable
    static int b; //static/template/class variable

    public void mA(){ //Instance method
        System.out.println("instance method");
    }
    public static void mB(){//static/class/template method
        System.out.println("static method");
    }
}
class Demo{
    public static void main(String args[]){
        Test.b=100; //Legal
        Test.mB(); //Legal

        Test t1=new Test();
        t1.a=200;
        t1.mA();
    }
}
```

```
Q43 Exercise
=====
class Test{
    int a;
    static int b;
}
class Demo{
    public static void main(String args[]){
        Test t1=new Test();
        Test t2=new Test();
        Test t3=new Test();
        t1.a=1;
        t1.b=2;
        t2.a=10;
        t2.b=20;
        t3.a=100;
        t3.b=200;

        System.out.println("t1 : "+t1.a+" "+t1.b);
        System.out.println("t2 : "+t2.a+" "+t2.b);
        System.out.println("t3 : "+t3.a+" "+t3.b);
    }
}
```



```
//////////  
Q44  
class Test{  
    int a;  
    static int b;  
}  
class Demo{  
    public static void main(String args[]){  
        //Test.a=100; //Illegal  
        Test.b=200; //Legal  
  
        Test t2=new Test();  
        t2.a=1000; //Legal  
        t2.b=2000; //Legal  
  
        Test t3=null;  
        t3.b=20000;  
        t3.a=10000; //Legal, Runtime Error (throws "NullPointerException)  
    }  
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q45 Static Methods

=====

```
class Test{
    int a;
    static int b;

    public void instanceMethod(){
        System.out.println("instance method");
    }
    public static void staticMethod(){
        System.out.println("static method");
    }
}

class Demo{
    public static void main(String args[]){
        Test.staticMethod();

        //Test.intanceMethod(); //Illegal
        Test t1=new Test();
        t1.instanceMethod();
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q46

```
class Test{
    int a;
    static int b;

    public void instanceMethod(){
        System.out.println(a);
        System.out.println(b);
        System.out.println(this);
    }
    public static void staticMethod(){
        System.out.println(a); //Illegal ?
        System.out.println(b);
        System.out.println(this); //Illegal
    }
}

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

//////////

Q47 Exercise

=====

```
class Date{  
    private int year;  
    private int month;  
    private int day;  
  
    Date(int year, int month, int day){  
        this.year=year;  
        this.month=month;  
        this.day=day;  
    }  
    public void printDate(){  
        System.out.println(year+"-"+month+"-"+day);  
    }  
}  
  
class Demo{  
    public static void main(String args[]){  
        Date d1=new Date(1999,12,31);  
        Date d2=new Date(1900,1,1);  
        Date d3=new Date(2024,5,13);  
  
        d1.printDate(); //1999-12-12  
        d2.printDate(); //1900-1-1  
        d3.printDate(); //2024-5-13  
    }  
}
```

//////////

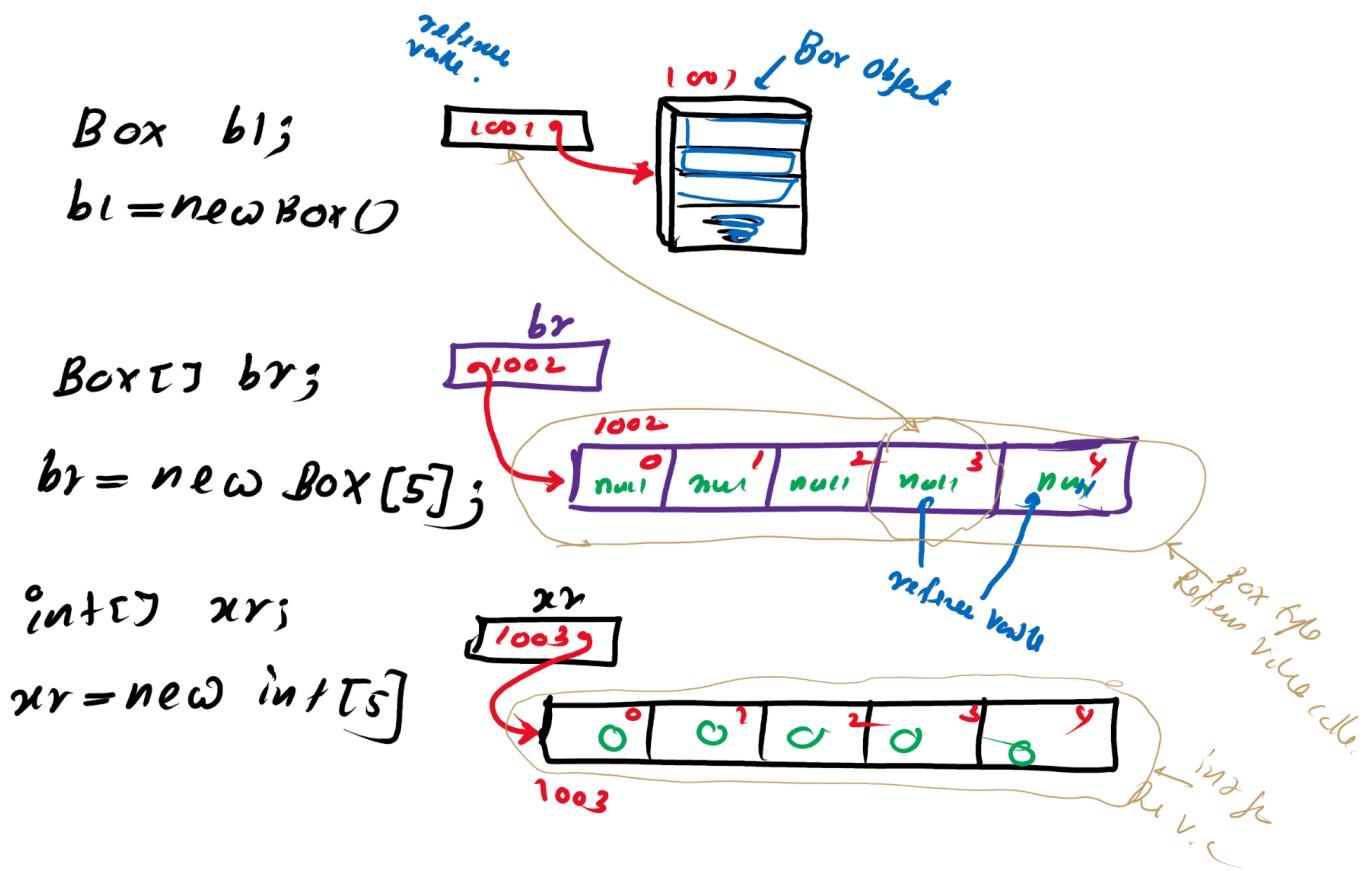
Q48 From Q47

=====

```
class Date{  
    private static int year;  
    private static int month;  
    private static int day;  
  
    Date(int year, int month, int day){  
        this.year=year;  
        this.month=month;  
        this.day=day;  
    }  
    public void printDate(){  
        System.out.println(year+"-"+month+"-"+day);  
    }  
}
```

```
class Demo{  
    public static void main(String args[]){  
        Date d1=new Date(1999,12,31);  
        Date d2=new Date(1900,1,1);  
        Date d3=new Date(2024,5,13);  
  
        d1.printDate(); //  
        d2.printDate(); //  
        d3.printDate(); //  
    }  
}
```

Object Arrays



```
Q49
class Box{
    int length;
    int width;
    int height;
    public Box(int length, int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
}
```

```

public void printVolume(){
    int volume;
    volume=length*width*height;
    System.out.println("Volume : "+volume);
}
}

class Demo{
    public static void main(String args[]){
        int[] xr=new int[5];
        xr[0]=100;

        Box[] br=new Box[5];
        br[0]=new Box(1,2,3);
        br[1]=new Box(10,20,30);
        br[2]=new Box(4,7,2);
        br[3]=new Box(8,3,1);
        br[4]=new Box(9,4,5);

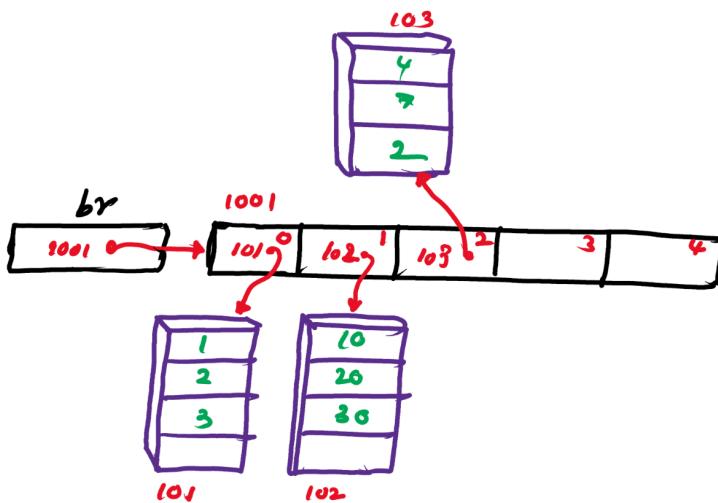
    }
}

```

```

Box[] br=new Box[5];
br[0]=new Box(1,2,3);
br[1]=new Box(10,20,30);
br[2]=new Box(4,7,2);
br[3]=new Box(8,3,1);
br[4]=new Box(9,4,5);

```



```

///////////////////////////////
Q50
class Box{
    int length;
    int width;
    int height;
    public Box(int length, int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}

```

```

class Demo{
    public static void main(String args[]){
        int[] xr={10,20,30,40,50};

        Box[] br={      new Box(1,2,3),
                        new Box(10,20,30),
                        new Box(4,7,2),
                        new Box(8,3,1),
                        new Box(9,4,5)
                };

    }

}

///////////////////////////////
Q51
class Box{
    int length;
    int width;
    int height;
    public Box(int length, int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}
class Demo{
    public static void main(String args[]){
        int[] xr={10,20,30,40,50};

        Box[] br={      new Box(1,2,3),
                        new Box(10,20,30),
                        new Box(4,7,2),
                        new Box(8,3,1),
                        new Box(9,4,5)
                };

        br[0].printVolume();
        br[1].printVolume();
        br[2].printVolume();
        br[3].printVolume();
        br[4].printVolume();

    }
}

```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q52

```
class Box{
    int length;
    int width;
    int height;
    public Box(int length, int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}
class Demo{
    public static void main(String args[]){
        int[] xr={10,20,30,40,50};

        Box[] br={      new Box(1,2,3),
                        new Box(10,20,30),
                        new Box(4,7,2),
                        new Box(8,3,1),
                        new Box(9,4,5)
                    };
        for(int i=0; i<br.length; i++){
            br[i].printVolume();
        }
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q53

```
class Box{
    int length;
    int width;
    int height;
    public Box(int length, int width, int height){
        this.length=length;
        this.width=width;
        this.height=height;
    }
    public void printVolume(){
        int volume;
        volume=length*width*height;
        System.out.println("Volume : "+volume);
    }
}
```

```

class Demo{
    public static void main(String args[]){
        int[] xr={10,20,30,40,50};

        Box[] br={      new Box(1,2,3),
                        new Box(10,20,30),
                        new Box(4,7,2),
                        new Box(8,3,1),
                        new Box(9,4,5)
                    };
        for(Box b1 : br){
            b1.printVolume();
        }

    }
}

```

//
Q54 Exercise

=====

```

class Order{
    String id; //customer id
    double amount;
    Order(String id, double amount){
        this.id=id;
        this.amount=amount;
    }
}
class Demo{
    public static void printOrders(Order[] orderArray){

    }
    public static void sortOrderArray(Order[] orderArray){

    }
    public static void main(String args[]){
        Order[] orderArray={
            new Order("C001",4500),
            new Order("C002",2500),
            new Order("C003",6500),
            new Order("C004",1500),
            new Order("C005",3500),
            new Order("C005",5500),
        };
        printOrders(orderArray);//[C001-4500, C002-2500, C003-6500,C004-1500,C005-3500,C005-5500]

        sortOrderArray(orderArray);

        printOrders(orderArray);//[C001-1500,C002-2500,C005-3500,C001-4500,C005-5500,C003-6500]
    }
}

```

```
////////////////////////////////////////////////////////////////////////
Q55 From Q54
=====
class Order{
    String id; //customer id
    double amount;
    Order(String id, double amount){
        this.id=id;
        this.amount=amount;
    }
}
class Demo{
    public static void printOrders(Order[] orderArray){
        System.out.print("[");
        for(Order order :orderArray){
            System.out.print(order.id+"-"+order.amount+", ");
        }
        System.out.println("\b\b]");
    }
    public static void sortOrderArray(Order[] orderArray){
        for(int i=orderArray.length-1; i>0; i--){
            for(int j=0; j<orderArray.length-1; j++){
                if(orderArray[j].amount>orderArray[j+1].amount){
                    Order temp=orderArray[j];
                    orderArray[j]=orderArray[j+1];
                    orderArray[j+1]=temp;
                }
            }
        }
    }
    public static void main(String args[]){
        Order[] orderArray={
            new Order("C001",4500),
            new Order("C002",2500),
            new Order("C003",6500),
            new Order("C004",1500),
            new Order("C005",3500),
            new Order("C005",5500),
        };
        printOrders(orderArray);//[C001-4500, C002-2500, C003-6500,C004-1500,C005-3500,C005-5500]

        sortOrderArray(orderArray);

        printOrders(orderArray);//[C001-1500,C002-2500,C005-3500,C001-4500,C005-5500,C003-6500]
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q56 Exercise
=====
class Order{
    private String id; //customer id
    private double amount;
    Order(String id, double amount){
        this.id=id;
        this.amount=amount;
    }
}
class Demo{
    public static void printOrders(Order[] orderArray){
        System.out.print("[");
        for(Order order :orderArray){
            System.out.print(order.id+"-"+order.amount+", ");
        }
        System.out.println("\b\b]");
    }
    public static void sortOrderArray(Order[] orderArray){
        for(int i=orderArray.length-1; i>0; i--){
            for(int j=0; j<orderArray.length-1; j++){
                if(orderArray[j].amount>orderArray[j+1].amount){
                    Order temp=orderArray[j];
                    orderArray[j]=orderArray[j+1];
                    orderArray[j+1]=temp;
                }
            }
        }
    }
    public static void main(String args[]){
        Order[] orderArray={
            new Order("C001",4500),
            new Order("C002",2500),
            new Order("C003",6500),
            new Order("C004",1500),
            new Order("C005",3500),
            new Order("C005",5500),
        };
        printOrders(orderArray);//[C001-4500, C002-2500, C003-6500,C004-1500,C005-3500,C005-5500]

        sortOrderArray(orderArray);

        printOrders(orderArray);//[C001-1500,C002-2500,C005-3500,C001-4500,C005-5500,C003-6500]
    }
}
```

Data Structures (Stack, Queue and List)

Q57 Step I

=====

```
class Demo{  
    public static void main(String args[]){  
        Stack s1=new Stack();  
        s1.push(10);  
        s1.push(20);  
        s1.push(30);  
        s1.push(40);  
        s1.push(50);  
        s1.printStack(); // [50,40,30,20,10]  
  
        s1.pop();  
        s1.printStack(); // [40,30,20,10]  
    }  
}
```

Q58 Step II

=====

```
class Stack{  
    public void push(int data){  
  
    }  
    public void printStack(){  
  
    }  
    public void pop(){  
  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Stack s1=new Stack();  
        s1.push(10);  
        s1.push(20);  
        s1.push(30);  
        s1.push(40);  
        s1.push(50);  
        s1.printStack(); // [50,40,30,20,10]  
  
        s1.pop();  
        s1.printStack(); // [40,30,20,10]  
    }  
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q59 Step III

=====

```
class Stack{  
    private int[] dataArray;  
    Stack(){  
        dataArray=new int[100];  
    }  
    public void push(int data){  
    }  
    public void printStack(){  
    }  
    public void pop(){  
    }  
}
```

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

```
        Stack s1=new Stack();  
        s1.push(10);  
        s1.push(20);  
        s1.push(30);  
        s1.push(40);  
        s1.push(50);  
        s1.printStack(); // [50,40,30,20,10]
```

```
        s1.pop();  
        s1.printStack(); // [40,30,20,10]
```

```
}
```

```
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q60 Step IV

=====

```
class Stack{  
    private int[] dataArray;
```

```
    private int nextIndex;
```

```
    Stack(){
```

```
        dataArray=new int[100];  
        nextIndex=0;
```

```
}
```

```
    public void push(int data){  
        dataArray[nextIndex++]=data;
```

```
}
```

```
    public void printStack(){
```

```
        System.out.print("[");  
        for(int i=nextIndex-1; i>=0; i--){  
            System.out.print(dataArray[i]+", ");
```

```
}
```

```
        System.out.println(nextIndex==0 ? "empty]:""\b\b]");
```

```
}
```

```

public void pop(){

}

}

class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack() // [50,40,30,20,10]

        s1.pop();
        s1.printStack() // [40,30,20,10]
    }
}

```

//////////
Q61 Step V

```

=====
class Stack{
    private int[] dataArray;
    private int nextIndex;
    Stack(){
        dataArray=new int[100];
        nextIndex=0;
    }
    public void push(int data){
        dataArray[nextIndex++]=data;
    }
    public void printStack(){
        System.out.print("[");
        for(int i=nextIndex-1; i>=0; i--){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println(nextIndex==0 ? "empty":"\b\b]");
    }
    public void pop(){
        nextIndex--;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
    }
}
```

```
s1.printStack(); ///[50,40,30,20,10]
s1.pop();
s1.printStack(); ///[40,30,20,10]
}
}
```

//////////

Q62 Step VI

=====

```
class Stack{
    private int[] dataArray;
    private int nextIndex;
    Stack(){
        dataArray=new int[100];
        nextIndex=0;
    }
    public void push(int data){
        dataArray[nextIndex++]=data;
    }
    public void printStack(){
        System.out.print("[");
        for(int i=nextIndex-1; i>=0; i--){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println(nextIndex==0 ? "empty":"\b\b]");
    }
    public void pop(){
        nextIndex--;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.printStack(); ///[empty]
        System.out.println("Size : "+s1.size()); //0
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); ///[50,40,30,20,10]
        System.out.println("Size : "+s1.size()); //5

        s1.pop();
        s1.printStack(); ///[40,30,20,10]
        System.out.println("Size : "+s1.size()); //4

        s1.clear();
        System.out.println("Size : "+s1.size()); //0
        s1.printStack(); ///[empty]
    }
}
```

```
////////////////////////////////////////////////////////////////////////
```

Q63 From Q62

=====

```
class Stack{  
    private int[] dataArray;  
    private int nextIndex;  
    Stack(){  
        dataArray=new int[100];  
        nextIndex=0;  
    }  
    public void push(int data){  
        dataArray[nextIndex++]=data;  
    }  
    public void printStack(){  
        System.out.print("[");  
        for(int i=nextIndex-1; i>=0; i--){  
            System.out.print(dataArray[i]+", ");  
        }  
        System.out.println(nextIndex==0 ? "empty]:""\b\b]");  
    }  
    public void pop(){  
        nextIndex--;  
    }  
    public int size(){  
        return nextIndex;  
    }  
    public void clear(){  
        nextIndex=0;  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Stack s1=new Stack();  
        s1.printStack(); // [empty]  
        System.out.println("Size : "+s1.size()); // 0  
        s1.push(10);  
        s1.push(20);  
        s1.push(30);  
        s1.push(40);  
        s1.push(50);  
        s1.printStack(); // [50,40,30,20,10]  
        System.out.println("Size : "+s1.size()); // 5  
  
        s1.pop();  
        s1.printStack(); // [40,30,20,10]  
        System.out.println("Size : "+s1.size()); // 4  
        s1.clear();  
        System.out.println("Size : "+s1.size()); // 0  
        s1.printStack(); // [empty]  
    }  
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q64 Step VII

=====

class Stack{

```
    private int[] dataArray;
    private int nextIndex;
    Stack(int capacity){
        dataArray=new int[capacity];
        nextIndex=0;
    }
    private boolean isFull(){
        return nextIndex>=dataArray.length;
    }
    private void extendsArray(){
        int[] tempdataArray=new int[dataArray.length*2];
        for (int i = 0; i < dataArray.length; i++){
            tempdataArray[i]=dataArray[i];
        }
        dataArray=tempdataArray;
    }
    public void push(int data){
        if(isFull()){
            extendsArray();
        }else{
            dataArray[nextIndex++]=data;
        }
    }
    public void printStack(){
        System.out.print("[");
        for(int i=nextIndex-1; i>=0; i--){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println(nextIndex==0 ? "empty]":"\b\b]");
    }
    private boolean isEmpty(){
        return nextIndex<=0;
    }
    public void pop(){
        ifisEmpty()){
            System.out.println("Stack is empty.. ");
        }else{
            nextIndex--;
        }
    }
    public int size(){
        return nextIndex;
    }
    public void clear(){
        nextIndex=0;
    }
}
```

```

class Demo{
    public static void main(String args[]){
        Stack s1=new Stack(5);
        s1.printStack(); // [empty]
        System.out.println("Size : "+s1.size()); // 0
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println("Size : "+s1.size()); // 5

        s1.push(60);
        s1.printStack(); // [60,50,40,30,20,10]

        s1.push(70);
        s1.printStack(); // [70,60,50,40,30,20,10]

        s1.clear();
        s1.pop(); // Stack is empty..
    }
}

```

//////////
Q65 Step VIII
=====

```

class Stack{
    private int[] dataArray;
    Stack(){
        dataArray=new int[0];
    }
    private void extendsArray(){
        int[] tempdataArray=new int[dataArray.length+1];
        for (int i = 0; i < dataArray.length; i++){
            tempdataArray[i]=dataArray[i];
        }
        dataArray=tempdataArray;
    }
    public void push(int data){
        extendsArray();
        dataArray[dataArray.length-1]=data;
    }
    public void printStack(){
        System.out.print("[");
        for(int i=dataArray.length-1; i>=0; i--){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println(isEmpty() ? "empty]":"\b\b]");
    }
}

```

```
private boolean isEmpty(){
    return dataArray.length==0;
}
public void pop(){
    if(isEmpty()){
        System.out.println("Stack is empty..");
    }else{
        int[] tempdataArray=new int[dataArray.length-1];
        for (int i = 0; i < tempdataArray.length; i++) {
            tempdataArray[i]=dataArray[i];
        }
        dataArray=tempdataArray;
    }
}
public int size(){
    return dataArray.length;
}
public void clear(){
    dataArray=new int[0];
}
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.printStack(); // [empty]
        System.out.println("Size : "+s1.size()); // 0
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println("Size : "+s1.size()); // 5

        s1.push(60);
        s1.printStack(); // [60,50,40,30,20,10]

        s1.push(70);
        s1.printStack(); // [70,60,50,40,30,20,10]

        s1.clear();
        s1.pop(); // Stack is empty..
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q66 From Q65

=====

class Stack{

 private int[] dataArray;

 Stack(){

 dataArray=new int[0];

 }

 private void extendsArray(){

 int[] tempdataArray=new int[dataArray.length+1];

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

 tempdataArray[i+1]=dataArray[i];

 }

 dataArray=tempdataArray;

 }

 public void push(int data){

 extendsArray();

 dataArray[0]=data;

 }

 public void printStack(){

 System.out.print("[");

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

 System.out.print(dataArray[i]+", ");

 }

 System.out.println(isEmpty() ? "empty]":"\b\b]");

 }

 private boolean isEmpty(){

 return dataArray.length==0;

 }

 public void pop(){

 if(isEmpty()){

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

 }else{

 int[] tempdataArray=new int[dataArray.length-1];

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

 tempdataArray[i]=dataArray[i+1];

 }

 dataArray=tempdataArray;

 }

 }

 public int size(){

 return dataArray.length;

 }

 public void clear(){

 dataArray=new int[0];

 }

}

```
class Demo{  
    public static void main(String args[]){  
        Stack s1=new Stack();  
        s1.printStack(); //#[empty]  
        System.out.println("Size : "+s1.size()); //0  
        s1.push(10);  
        s1.push(20);  
        s1.push(30);  
        s1.push(40);  
        s1.push(50);  
        s1.printStack(); //#[50,40,30,20,10]  
        System.out.println("Size : "+s1.size()); //5  
  
        s1.pop();  
        s1.printStack(); //#[50,40,30,20,10]  
    }  
}
```

//////////

Class “Queue”

//////////

Q67 Step I

=====

```
class Demo{  
    public static void main(String args[]){  
        Queue q1=new Queue();  
        q1.enQueue(100);  
        q1.enQueue(200);  
        q1.enQueue(300);  
        q1.enQueue(400);  
        q1.enQueue(500);  
        q1.printQueue(); //#[100,200,300,400,500]  
  
        q1.deQueue();  
        q1.printQueue(); //#[200,300,400,500]  
    }  
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q68 Step II

=====

class Queue{

 private int[] dataArray;

 private int nextIndex;

 public Queue(int capacity){

 dataArray=new int[capacity];

 nextIndex=0;

 }

 public void enQueue(int data){

 }

 public void deQueue(){

 }

 public void printQueue(){

 }

}

class Demo{

 public static void main(String args[]){

 Queue q1=new Queue(10);

 q1.enQueue(100);

 q1.enQueue(200);

 q1.enQueue(300);

 q1.enQueue(400);

 q1.enQueue(500);

 q1.printQueue(); // [100, 200, 300, 400, 500]

 q1.deQueue();

 q1.printQueue(); // [200, 300, 400, 500]

 }

}

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q69 Step III

=====

class Queue{

 private int[] dataArray;

 private int nextIndex;

 public Queue(int capacity){

 dataArray=new int[capacity];

 nextIndex=0;

 }

 private void extendsArray(){

 int[] tempArray=new int[dataArray.length*2];

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

 tempArray[i]=dataArray[i];

 }

 dataArray=tempArray;

}

```
private boolean isFull(){
    return nextIndex>=dataArray.length;
}
public void enQueue(int data){
    if(isFull()){
        extendsArray();
    }
    dataArray[nextIndex++]=data;
}
public void deQueue(){
    if(!isEmpty()){
        for(int i=0; i<nextIndex-1; i++){
            dataArray[i]=dataArray[i+1];
        }
        nextIndex--;
    }
}
public boolean isEmpty(){
    return dataArray.length<=0;
}
public void printQueue(){
    System.out.print("[");
    for(int i=0; i<nextIndex; i++){
        System.out.print(dataArray[i]+", ");
    }
    System.out.println(isEmpty() ? "empty]":"\b\b]");
}

}
class Demo{
    public static void main(String args[]){
        Queue q1=new Queue(10);
        q1.enQueue(100);
        q1.enQueue(200);
        q1.enQueue(300);
        q1.enQueue(400);
        q1.enQueue(500);
        q1.printQueue() // [100,200,300,400,500]

        q1.deQueue();
        q1.printQueue() // [200,300,400,500]
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q70 step IV

=====

```
class Queue{  
    private int[] dataArray;  
    private int nextIndex;  
    private int intCapacity;  
  
    public Queue(int intCapacity){  
        dataArray=new int[intCapacity];  
        nextIndex=0;  
        this.intCapacity=intCapacity;  
    }  
    private void extendsArray(){  
        int[] tempArray=new int[dataArray.length*2];  
        for (int i = 0; i < dataArray.length; i++){  
            tempArray[i]=dataArray[i];  
        }  
        dataArray=tempArray;  
    }  
    private boolean isFull(){  
        return nextIndex>=dataArray.length;  
    }  
    public void enQueue(int data){  
        if(isFull()){  
            extendsArray();  
        }  
        dataArray[nextIndex++]=data;  
    }  
    public void deQueue(){  
        if(!isEmpty()){  
            for(int i=0; i<nextIndex-1; i++){  
                dataArray[i]=dataArray[i+1];  
            }  
            nextIndex--;  
        }  
    }  
    public int peek(){  
        return isEmpty()?-1:dataArray[0];  
    }  
    public int poll(){  
        return -1;  
    }  
    public int search(int data){  
        return -1;  
    }  
    public boolean isEmpty(){  
        return dataArray.length<=0;  
    }  
}
```

```

public void printQueue(){
    System.out.print("[");
    for(int i=0; i<nextIndex; i++){
        System.out.print(dataArray[i]+", ");
    }
    System.out.println(isEmpty() ? "empty]":"\b\b]");
}

public int size(){
    return nextIndex;
}

public void clear(){
    nextIndex=0;
    dataArray=new int[intCapacity];
}

}

class Demo{
    public static void main(String args[]){
        Queue q1=new Queue(10);
        q1.enQueue(100);
        q1.enQueue(200);
        q1.enQueue(300);
        q1.enQueue(400);
        q1.enQueue(500);
        q1.printQueue() // [100,200,300,400,500]

        q1.deQueue();
        q1.printQueue() // [200,300,400,500]
    }
}

```

//
Class “List” (Insertion order, Random access)

//
Q71
class List{

```

private int[] dataArray;
private int nextIndex;

public List(int intCapacity){
    dataArray=new int[intCapacity];
    nextIndex=0;
}

public void add(int data){ //add to the last
    //if(isFull()){extendsArray();}
    dataArray[nextIndex++]=data;
}

```

```

public void printList(){
    System.out.print("[");
    for (int i = 0; i < nextIndex; i++){
        System.out.print(dataArray[i]+", ");
    }
    System.out.println("\b\b]");
}
}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        list.printList(); //#[10,20,30,40,50]
    }
}

```

//////////
Q72 Step II
=====

```

class List{
    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];
        nextIndex=0;
    }

    public void add(int data){ //add to the last
        //if(isFull()){extendsArray();}
        dataArray[nextIndex++]=data;
    }

    public void printList(){
        System.out.print("[");
        for (int i = 0; i < nextIndex; i++){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println("\b\b]");
    }
}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
    }
}

```

```

list.printList(); // [10,20,30,40,50]

list.add(2,22); // add(index, data);
list.printList(); // [10,20,22,30,40,50]

list.add(6,60); // add(index, data);
list.printList(); // [10,20,22,30,40,50,60]
}

}

```

//////////

Q73 From Q72

=====

```

class List{
    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];
        nextIndex=0;
    }
    public void add(int data){ // add to the last
        // if(isFull()) {extendsArray();}
        dataArray[nextIndex++]=data;
    }
    public void add(int index, int data){
        if(index>=0 && index<=nextIndex){
            // if(isFull()) {extendsArray();}
            for(int i=nextIndex; i>index;i--){
                dataArray[i]=dataArray[i-1];
            }
            dataArray[index]=data;
            nextIndex++;
        }
    }
    public void printList(){
        System.out.print("[");
        for (int i = 0; i < nextIndex; i++){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println("\b\b]");
    }
}
class Demo{
    public static void main(String args[]){
        List list=new List(10);
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
    }
}
```

```

list.printList(); // [10,20,30,40,50]

list.add(2,22); // add(index, data);
list.printList(); // [10,20,22,30,40,50]

list.add(6,60); // add(index, data);
list.printList(); // [10,20,22,30,40,50,60]

list.add(0,100); // add(index, data);
list.printList(); // [100,20,22,30,40,50,60]
}

}

```

//////////

Q74 Step III

=====

```

class List{
    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];
        nextIndex=0;
    }

    public void add(int data){ // add to the last
        // if(isFull()){extendsArray();}
        dataArray[nextIndex++]=data;
    }

    public void add(int index, int data){
        if(index>=0 && index<=nextIndex){
            // if(isFull()){extendsArray();}
            for(int i=nextIndex; i>index;i--){
                dataArray[i]=dataArray[i-1];
            }
            dataArray[index]=data;
            nextIndex++;
        }
    }

    public void printList(){
        System.out.print("[");
        for (int i = 0; i < nextIndex; i++){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println("\b\b]");
    }
}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        list.add(10);
        list.add(20);
    }
}
```

```

        list.add(30);
        list.add(40);
        list.add(50);
        list.add(60);
        list.printList(); // [10,20,30,40,50,60]

        list.remove(2); //remove(index);
        list.printList(); // [10,20,40,50,60]
    }
}

```

//////////

Q75 From Q74

=====

```

class List{
    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];
        nextIndex=0;
    }

    public void add(int data){ //add to the last
        //if(isFull()){extendsArray();}
        dataArray[nextIndex++]=data;
    }

    public void add(int index, int data){
        if(index>=0 && index<=nextIndex){
            //if(isFull()){extendsArray();}
            for(int i=nextIndex; i>index;i--){
                dataArray[i]=dataArray[i-1];
            }
            dataArray[index]=data;
            nextIndex++;
        }
    }

    public void remove(int index){
        if(index>=0 && index<nextIndex){
            for(int i=index; i<nextIndex-1; i++){
                dataArray[i]=dataArray[i+1];
            }
            nextIndex--;
        }
    }

    public void printList(){
        System.out.print("[");
        for (int i = 0; i < nextIndex; i++){
            System.out.print(dataArray[i]+", ");
        }
        System.out.println("\b\b]");
    }
}
```

```

}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        list.add(60);
        list.printList(); //|[10,20,30,40,50,60]

        list.remove(2); //remove(index);
        list.printList(); //|[10,20,40,50,60]
    }
}

```

//////////

Q76

```

class List{
    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];
        nextIndex=0;
    }

    public void add(int data){ //add to the last
        //if(isFull())extendsArray();
        dataArray[nextIndex++]=data;
    }

    public void add(int index, int data){
        if(index>=0 && index<=nextIndex){
            //if(isFull())extendsArray();
            for(int i=nextIndex; i>index;i--){
                dataArray[i]=dataArray[i-1];
            }
            dataArray[index]=data;
            nextIndex++;
        }
    }

    public void remove(int index){
        if(index>=0 && index<nextIndex){
            for(int i=index; i<nextIndex-1; i++){
                dataArray[i]=dataArray[i+1];
            }
            nextIndex--;
        }
    }
}

```

```

public void printList(){
    System.out.print("[");
    for (int i = 0; i < nextIndex; i++){
        System.out.print(dataArray[i]+", ");
    }
    System.out.println("\b\b]");
}
}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        list.add(60);
        list.printList(); ///[10,20,30,40,50,60]

        int data=list.get(2);
        System.out.println("element of index 2 : "+data);

        data=list.get(7);
        System.out.println("element of index 8 : "+data); //-1
    }
}

```

//////////
Q77 From Q76
=====

```

class List{
    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];
        nextIndex=0;
    }

    public int get(int index){
        if(index>=0 && index<nextIndex){
            return dataArray[index];
        }
        return -1;
    }

    public void add(int data){ //add to the last
        //if(isFull())extendsArray();
        dataArray[nextIndex++]=data;
    }

    public void add(int index, int data){
        if(index>=0 && index<=nextIndex){
            //if(isFull())extendsArray();
        }
    }
}
```

```

        for(int i=nextIndex; i>index;i--){
            dataArray[i]=dataArray[i-1];
        }
        dataArray[index]=data;
        nextIndex++;
    }
}

public void remove(int index){
    if(index>=0 && index<nextIndex){
        for(int i=index; i<nextIndex-1; i++){
            dataArray[i]=dataArray[i+1];
        }
        nextIndex--;
    }
}

public void printList(){
    System.out.print("[");
    for (int i = 0; i < nextIndex; i++){
        System.out.print(dataArray[i]+", ");
    }
    System.out.println("\b\b]");
}
}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        list.add(60);
        list.printList(); ///[10,20,30,40,50,60]

        int data=list.get(2);
        System.out.println("element of index 2 : "+data);

        data=list.get(7);
        System.out.println("element of index 8 : "+data); //-1
    }
}

```

//////////////
Q78 Exercise

=====

class List{

```

    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];

```

```
    nextIndex=0;
}
public int get(int index){
    if(index>=0 && index<nextIndex){
        return dataArray[index];
    }
    return -1;
}
public void add(int data){ //add to the last
    //if(isFull()){extendsArray();}
    dataArray[nextIndex++]=data;
}
public void add(int index, int data){
    if(index>=0 && index<=nextIndex){
        //if(isFull()){extendsArray();}
        for(int i=nextIndex; i>index;i--){
            dataArray[i]=dataArray[i-1];
        }
        dataArray[index]=data;
        nextIndex++;
    }
}
public void remove(int index){
    if(index>=0 && index<nextIndex){
        for(int i=index; i<nextIndex-1; i++){
            dataArray[i]=dataArray[i+1];
        }
        nextIndex--;
    }
}
public void printList(){
    System.out.print("[");
    for (int i = 0; i < nextIndex; i++){
        System.out.print(dataArray[i]+", ");
    }
    System.out.println("\b\b]");
}
}
class Demo{
    public static void main(String args[]){
        List list=new List(10);
        System.out.println(list.toString()); // [empty]
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        list.add(60);
        System.out.println(list.toString());// [10,20,30,40,50,60]
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q79 From Q78

=====

```
class List{
    private int[] dataArray;
    private int nextIndex;

    public List(int intCapacity){
        dataArray=new int[intCapacity];
        nextIndex=0;
    }

    public int get(int index){
        if(index>=0 && index<nextIndex){
            return dataArray[index];
        }
        return -1;
    }

    public void add(int data){ //add to the last
        //if(isFull())extendsArray();
        dataArray[nextIndex++]=data;
    }

    public void add(int index, int data){
        if(index>=0 && index<=nextIndex){
            //if(isFull())extendsArray();
            for(int i=nextIndex; i>index;i--){
                dataArray[i]=dataArray[i-1];
            }
            dataArray[index]=data;
            nextIndex++;
        }
    }

    public void remove(int index){
        if(index>=0 && index<nextIndex){
            for(int i=index; i<nextIndex-1; i++){
                dataArray[i]=dataArray[i+1];
            }
            nextIndex--;
        }
    }

    public void printList(){
        //System.out.println(this);
        System.out.println(toString());
    }

    public String toString(){
        String s="[";
        for (int i = 0; i < nextIndex; i++){
            s+=dataArray[i]+", ";
        }
        return isEmpty() ? "[empty]":(s+="\b\b");
    }
}
```

```
public boolean isEmpty(){
    return nextIndex==0;
}
}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        System.out.println(list.toString()); // [empty]
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        list.add(60);
        System.out.println(list); // [10, 20, 30, 40, 50, 60]

        list.printList();
    }
}
```

//////////
Q80
class Box{

```
}
```

class Demo{
 public static void main(String args[]){
 Box b1=new Box();
 System.out.println(b1); // b1.toString()
 System.out.println(b1.toString());

 }
}

//////////
Q81 From Q80
=====

```
class Box{
    public String toString(){
        return "Niroth";
    }
}

class Demo{
    public static void main(String args[]){
        Box b1=new Box();
        System.out.println(b1); // Niroth
        System.out.println(b1.toString());

    }
}
```

//

Q82

```
class List{  
    private int[] dataArray;  
    private int nextIndex;  
  
    public List(int intCapacity){  
        dataArray=new int[intCapacity];  
        nextIndex=0;  
    }  
    public int get(int index){  
        if(index>=0 && index<nextIndex){  
            return dataArray[index];  
        }  
        return -1;  
    }  
    public void add(int data){ //add to the last  
        //if(isFull()){extendsArray();}  
        dataArray[nextIndex++]=data;  
    }  
    public void add(int index, int data){  
        if(index>=0 && index<=nextIndex){  
            //if(isFull()){extendsArray();}  
            for(int i=nextIndex; i>index;i--){  
                dataArray[i]=dataArray[i-1];  
            }  
            dataArray[index]=data;  
            nextIndex++;  
        }  
    }  
    public void remove(int index){  
        if(index>=0 && index<nextIndex){  
            for(int i=index; i<nextIndex-1; i++){  
                dataArray[i]=dataArray[i+1];  
            }  
            nextIndex--;  
        }  
    }  
    public void printList(){  
        //System.out.println(this);  
        System.out.println(toString());  
    }  
    public String toString(){  
        String s="[";  
        for (int i = 0; i < nextIndex; i++){  
            s+=dataArray[i]+", ";  
        }  
        return isEmpty() ? "[empty]":(s+="\b\b");  
    }  
}
```

```

public boolean isEmpty(){
    return nextIndex==0;
}
}

class Demo{
    public static void main(String args[]){
        List list=new List(10);
        System.out.println(list.toString()); // [empty]
        list.add(10);
        list.add(20);
        list.add(30);
        list.add(40);
        list.add(50);
        list.add(60);
        System.out.println(list); // [10, 20, 30, 40, 50, 60]

        int[] ar=new int[]{21, 22, 23};
        list.add(2, ar);
        System.out.println(list); // [10, 20, 21, 22, 23, 30, 40, 50, 60]

        list.remove(2, 4); // remove(startIndex, endIndex)
        System.out.println(list); // [10, 20, 30, 40, 50, 60]

        int[] dataArray=list.toArray();
        System.out.println(Arrays.toString(dataArray)); // [10, 20, 30, 40, 50, 60]
    }
}

```

//////////////////////////////

Q83 Exercise

=====

```

class Customer{
    private String id;
    private double salary;
    Customer(String id, double salary){
        this.id=id;
        this.salary=salary;
    }
    public void setId(String id){
        this.id=id;
    }
    public void setSalary(double salary){
        this.salary=salary;
    }
}

```

```
class CustomerStack{
    private Customer[] customerArray;
    private int nextIndex;

    CustomerStack(int intCapacity){
        customerArray=new Customer[intCapacity];
        nextIndex=0;
    }
    public void add(Customer customer){

    }
    public void add(int index, Customer customer){

    }
    public void remove(int index){

    }
    public Customer get(int index){

    }
}

class Demo{
    public static void main(String args[]){
        CustomerStack custStack=new CustomerStack(100);
        custStack.push(new Customer("C001",15000));
        custStack.push(new Customer("C002",65000));
        custStack.push(new Customer("C003",25000));
        custStack.push(new Customer("C004",45000));
        System.out.println(custStack);//[C004-45000, C003-25000, C002-65000, C001-15000]

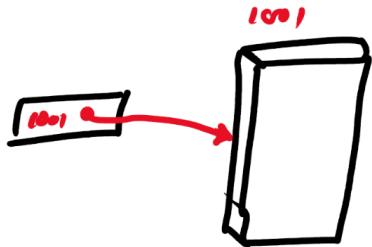
        custStack.pop();
        System.out.println(custStack);//[C003-25000, C002-65000, C001-15000]
    }
}
```

Data Structures using “Linked List”- Nodes

Q84 Step 1

=====

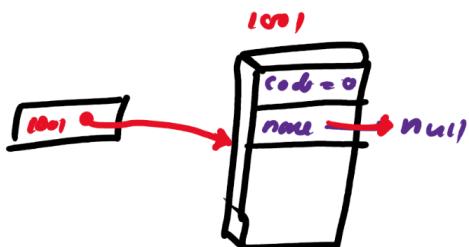
```
class Customer{  
}  
  
class Demo{  
    public static void main(String args[]){  
        Customer c1;  
        c1=new Customer();  
    }  
}
```



Q85 Step II

=====

```
class Customer{  
    int code;  
    String name;  
}  
  
class Demo{  
    public static void main(String args[]){  
        Customer c1;  
        c1=new Customer();  
    }  
}
```

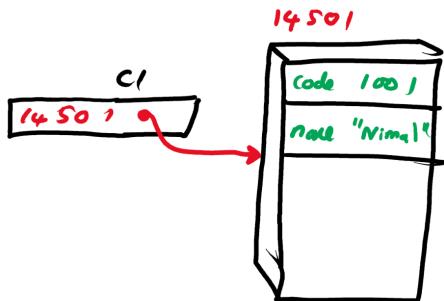


||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q86 Step III

=====

```
class Customer{  
    int code;  
    String name;  
    Customer(int code, String name){  
        this.code=code;  
        this.name=name;  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Customer c1;  
        c1=new Customer(1001,"Nimal");  
    }  
}
```

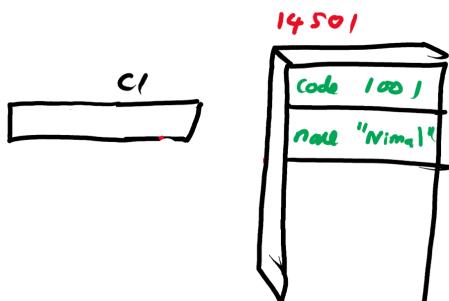


||||||||||||||||||||||||||||||||||||||||||||||||||||

Q87 Step IV

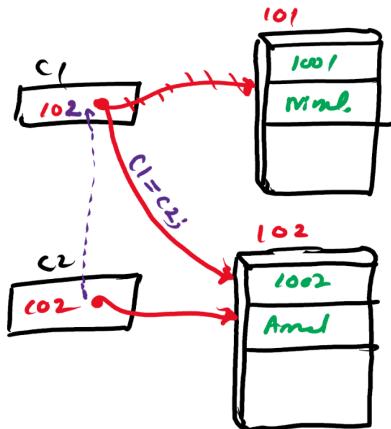
=====

```
class Customer{  
    int code;  
    String name;  
    Customer(int code, String name){  
        this.code=code;  
        this.name=name;  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Customer c1;  
        new Customer(1001,"Nimal");  
    }  
}
```



```
///////////////////////////////
Q88 Step V
=====
class Customer{
    int code;
    String name;
    Customer(int code, String name){
        this.code=code;
        this.name=name;
    }
}
class Demo{
    public static void main(String args[]){
        Customer c1=new Customer(1001,"Nimal");
        Customer c2=new Customer(1002,"Amal");

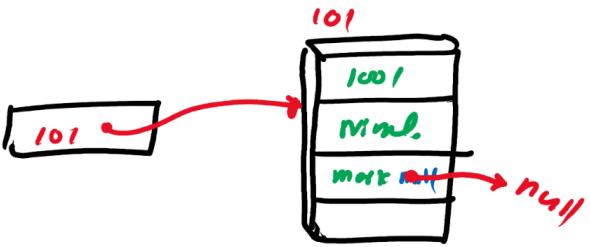
        c1=c2;
    }
}
```



```
///////////////////////////////
Q89
class Marks{

}
class Student{
    int code;
    String name;
    Marks marks;

    Student(int code, String name){
        this.code=code;
        this.name=name;
    }
}
class Demo{
    public static void main(String args[]){
        Student s1=new Student(1001,"Nimal");
    }
}
```



//////////

Q90 Step VI

=====

class Marks{

}

class Student{

 int code;

 String name;

 Marks marks;

 Student(int code, String name){

 this.code=code;

 this.name=name;

 }

}

class Demo{

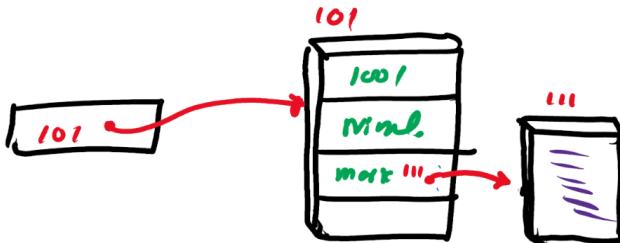
 public static void main(String args[]){

 Student s1=new Student(1001,"Nimal");

 s1.marks=new Marks();

 }

}



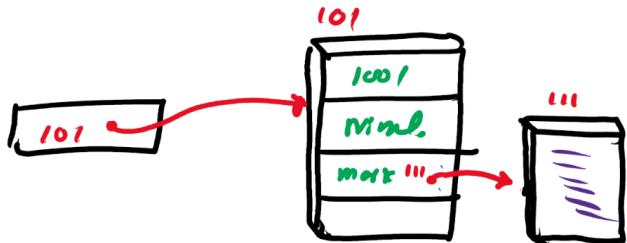
```
///////////////////////////////
Q91 Step VII
=====
class Marks{

}

class Student{
    int code;
    String name;
    Marks marks;

    Student(int code, String name, Marks marks){
        this.code=code;
        this.name=name;
        this.marks=marks;
    }
}

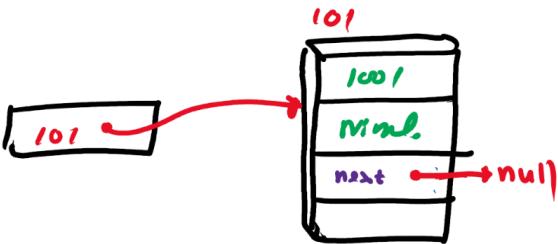
class Demo{
    public static void main(String args[]){
        Student s1=new Student(1001,"Nimal",new Marks());
    }
}
```



```
///////////////////////////////
Q92 Step VIII
=====
class Student{
    int code;
    String name;
    Student next;

    Student(int code, String name){
        this.code=code;
        this.name=name;
    }
}

class Demo{
    public static void main(String args[]){
        Student s1=new Student(1001,"Nimal");
    }
}
```



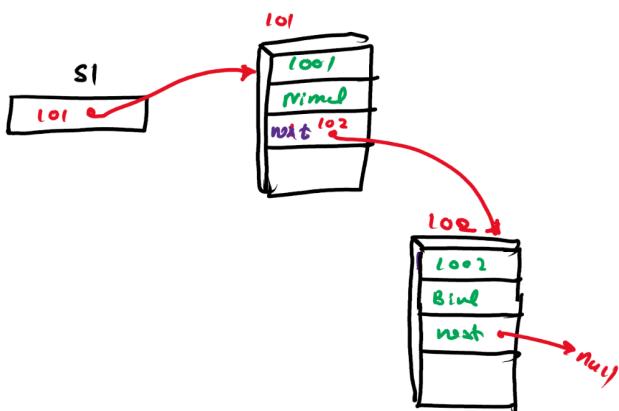
//////////
Q93 Step IX

=====

```
class Student{
    int code;
    String name;
    Student next;
```

```
Student(int code, String name){
    this.code=code;
    this.name=name;
}
}

class Demo{
    public static void main(String args[]){
        Student s1=new Student(1001,"Nimal");
        s1.next=new Student(1002,"Bimal");
    }
}
```



//////////
Q94 Step X

=====

```
class Student{
    int code;
    String name;
    Student next;
```

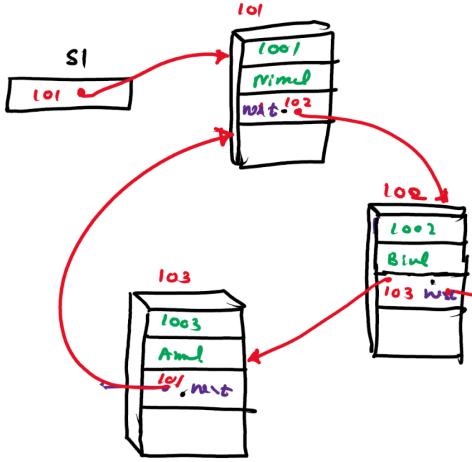


```
Student(int code, String name){
    this.code=code;
    this.name=name;
}
}
```

```

class Demo{
    public static void main(String args[]){
        Student s1=new Student(1001,"Nimal");
        s1.next=new Student(1002,"Bimal");
        s1.next.next=new Student(1003,"Aimal");
        s1.next.next.next=s1;
    }
}

```



Class “Stack”

Q95 Step 1

=====

```
class Stack{
```

}

```
class Demo{
```

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

```
        Stack s1=new Stack();
```

}

}

Q96 Step II

=====

```
class Stack{
```

}

```
class Node{
```

}

```
class Demo{
```

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

```
        Stack s1=new Stack();
```

}

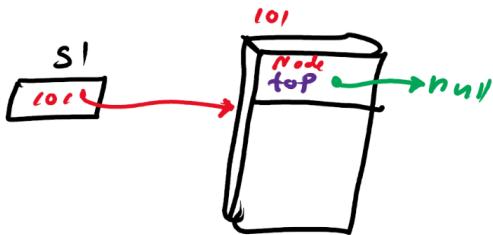
}

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q97 Step III

=====

```
class Stack{  
    Node top;  
}  
class Node{  
}  
class Demo{  
    public static void main(String args[]){  
        Stack s1=new Stack();  
    }  
}
```

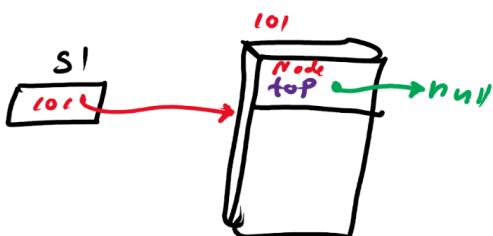


||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q98 Step IV

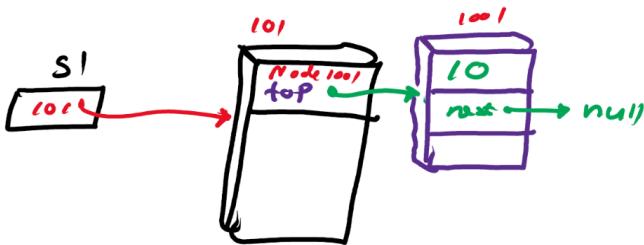
=====

```
class Stack{  
    Node top;  
}  
class Node{  
    int data;  
    Node next;  
    Node(int data){  
        this.data=data;  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        Stack s1=new Stack();  
    }  
}
```



```
///////////////////////////////
Q99 Step V
=====
class Stack{
    Node top;

    public void push(int data){
        //
        //
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
    }
}
```



```
///////////////////////////////
Q100 From Q99
=====
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        top=n1;
        //top=new Node(data);
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
```

```
class Demo{  
    public static void main(String args[]){  
        Stack s1=new Stack();  
        s1.push(10);  
    }  
}
```

A decorative horizontal border consisting of a repeating pattern of diagonal hatching.

Q101 Step VI

====

```
class Stack{
```

Node top;

```
public void push(int data){
```

```
Node n1=new Node(data);
```

top=n1;

}

}

```
class Node{
```

```
int data;
```

Node next;

```
Node(int data){
```

```
this.data=data;
```

}

}

```
class Demo{
```

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

```
Stack s1=new Stack();
```

```
s1.push(10);
```

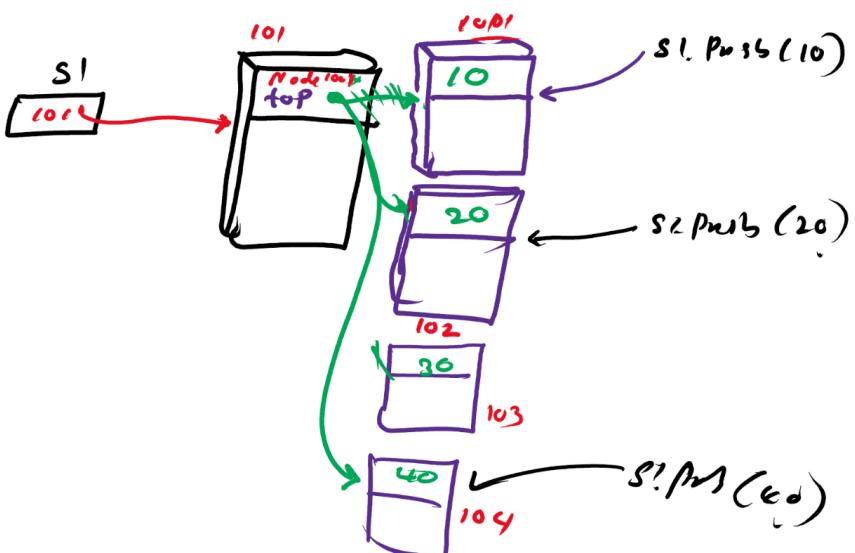
```
s1.push(20);
```

```
s1.push(30);
```

```
s1.push(40);
```

}

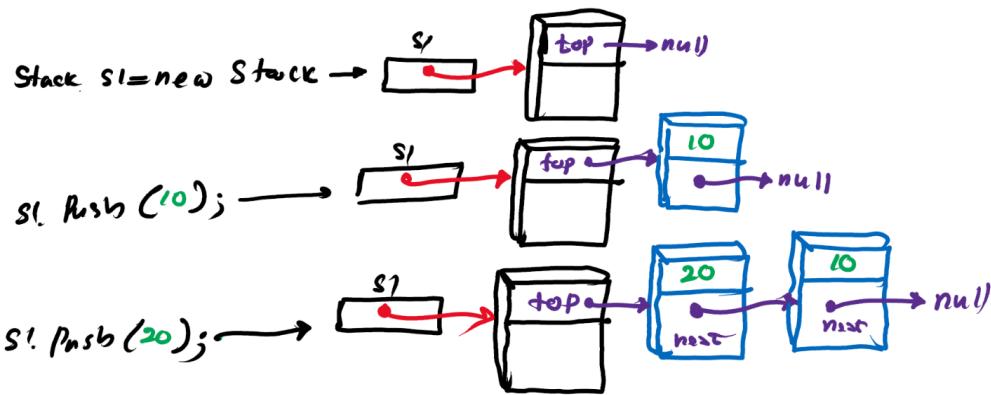
}



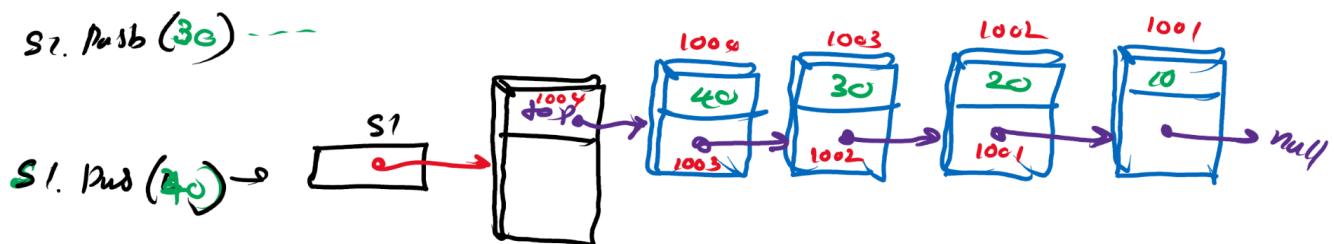
||||||||||||||||||||||||||||||||||||||||||||

Q102 Step VIII

=====



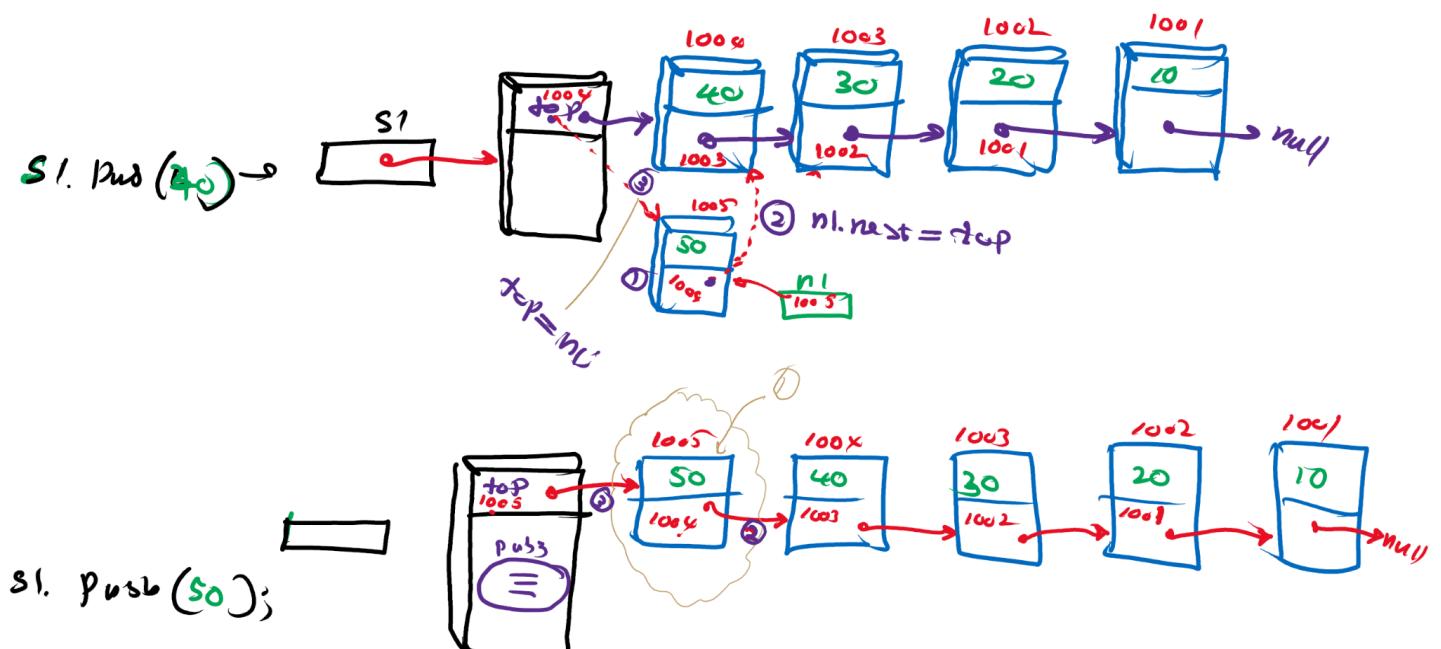
s1.push(30) ---



||||||||||||||||||||||||||||||||||||||||

Q103 From Q102

=====



```
///////////////////////////////
Q104 From Q103
=====
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
    }
}
```

```
/////////////////////////////
Q105 Step IX
=====
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public void printStack(){

    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
```

```
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); //|[50,40,30,20,10]
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q106 From Q105

=====

```
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public void printStack(){
        System.out.print("[");
        System.out.print(top.data+", ");
        System.out.print(top.next.data+", ");
        System.out.print(top.next.next.data+", ");
        System.out.println("\b\b]");
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); //|[50,40,30,20,10]
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q107 From Q106
=====
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public void printStack(){
        System.out.print("[");
        while(top!=null){
            System.out.print(top.data+", ");
            top=top.next;
        }
        System.out.println("\b\b]");
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        s1.printStack(); // []
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q108 From Q107
=====
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public void printStack(){
        System.out.print("[");
        Node temp=top;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println("\b\b]");
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        s1.printStack(); // [50,40,30,20,10]
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q109 Step X
=====
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public void printStack(){
        System.out.print("[");
        Node temp=top;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println("\b\b]");
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]

        s1.pop();
        s1.printStack(); // [40,30,20,10]
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q110 From Q109
=====
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public void printStack(){
        System.out.print("[");
        Node temp=top;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println("\b\b]");
    }
    public void pop(){
        top=top.next;
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]

        s1.pop();
        s1.printStack(); // [40,30,20,10]
    }
}
```

```
////////////////////////////////////////////////////////////////////////
```

Q111 Step X

=====

```
class Stack{
```

```
    Node top;
    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
```

```
    public void printStack(){
        System.out.print("[");
        Node temp=top;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println("\b\b]");
    }
```

```
    public void pop(){
        top=top.next;
    }
```

```
}
```

```
class Node{
```

```
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
```

```
}
```

```
class Demo{
```

```
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.printStack();//[empty]
        System.out.println("Size of the stack : "+s1.size()); //0
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println("Size of the stack : "+s1.size()); //5
```

```
        s1.pop();
        s1.printStack(); // [40,30,20,10]
        System.out.println("Size of the stack : "+s1.size()); //4
        s1.clear();
        s1.printStack();//[empty]
        System.out.println("Size of the stack : "+s1.size()); //0
```

```
}
```

```
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q112 From Q111

=====

class Stack{

 Node top;

 public void push(int data){

 Node n1=new Node(data);

 n1.next=top;

 top=n1;

 }

 public boolean isEmpty(){

 return top==null;

 }

 public int size(){

 Node temp=top;

 int count=0;

 while(temp!=null){

 temp=temp.next;

 count++;

 }

 return count;

 }

 public void printStack(){

 System.out.print("[");

 Node temp=top;

 while(temp!=null){

 System.out.print(temp.data+", ");

 temp=temp.next;

 }

 System.out.println(isEmpty()? "empty]":"\b\b]");

 }

 public void pop(){

 top=top.next;

 }

 public void clear(){

 top=null;

 }

}

class Node{

 int data;

 Node next;

 Node(int data){

 this.data=data;

 }

}

class Demo{

 public static void main(String args[]){

 Stack s1=new Stack();

 s1.printStack();//[empty]

 System.out.println("Size of the stack : "+s1.size()); //0

```

s1.push(10);
s1.push(20);
s1.push(30);
s1.push(40);
s1.push(50);
s1.printStack() // [50,40,30,20,10]
System.out.println("Size of the stack : "+s1.size()); // 5

s1.pop();
s1.printStack(); // [40,30,20,10]
System.out.println("Size of the stack : "+s1.size()); // 4

s1.clear();
s1.printStack(); // [empty]
System.out.println("Size of the stack : "+s1.size()); // 0
}

}

```

//////////
Q113 Step XI

=====

```

class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public boolean isEmpty(){
        return top==null;
    }
    public int size(){
        Node temp=top;
        int count=0;
        while(temp!=null){
            temp=temp.next;
            count++;
        }
        return count;
    }
    public void printStack(){
        System.out.print("[");
        Node temp=top;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println(isEmpty()? "empty]":"\b\b]");
    }
}

```

```
public void pop(){
    top=top.next;
}
public void clear(){
    top=null;
}
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println();

        int topData=s1.peek();
        System.out.println("Top Data : "+topData); // 50
        System.out.println("After calling peek()....");
        s1.printStack(); // [50,40,30,20,10]

        topData=s1.poll();
        System.out.println("Top Data : "+topData); // 50
        System.out.println("After calling poll()....");
        s1.printStack(); // [40,30,20,10]
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q114 From Q113

=====

class Stack{

 Node top;

 public void push(int data){

 Node n1=new Node(data);

 n1.next=top;

 top=n1;

 }

 public boolean isEmpty(){

 return top==null;

 }

 public int size(){

 Node temp=top;

 int count=0;

 while(temp!=null){

 temp=temp.next;

 count++;

 }

 return count;

 }

 public void printStack(){

 System.out.print("[");

 Node temp=top;

 while(temp!=null){

 System.out.print(temp.data+", ");

 temp=temp.next;

 }

 System.out.println(isEmpty() ? "empty]":"\b\b]");

 }

 public void pop(){

 top=top.next;

 }

 public void clear(){

 top=null;

 }

 public int peek(){

 return isEmpty() ? -1: top.data;

 }

 public int poll(){

 if(isEmpty()){

 return -1;
}

 }

 int data=top.data;

 pop();

 return data;

}

}

```

class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println();

        int topData=s1.peek();
        System.out.println("Top Data : "+topData); // 50
        System.out.println("After calling peek()....");
        s1.printStack(); // [50,40,30,20,10]
        System.out.println();

        topData=s1.poll();
        System.out.println("Top Data : "+topData); // 50
        System.out.println("After calling poll()....");
        s1.printStack(); // [40,30,20,10]
    }
}

```

//////////
Q115 Step XII
=====

```

import java.util.*;
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public boolean isEmpty(){
        return top==null;
    }
    public int size(){
        Node temp=top;
        int count=0;
        while(temp!=null){
            temp=temp.next;
            count++;
        }
        return count;
    }
}
```

```

        count++;
    }
    return count;
}
public void printStack(){
    System.out.print("[");
    Node temp=top;
    while(temp!=null){
        System.out.print(temp.data+", ");
        temp=temp.next;
    }
    System.out.println(isEmpty()? "empty]":"\b\b]");
}
public void pop(){
    top=top.next;
}
public void clear(){
    top=null;
}
public int peek(){
    return isEmpty() ? -1: top.data;
}
public int poll(){
    if(isEmpty()){
        return -1;
    }
    int data=top.data;
    pop();
    return data;
}
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println();

        int index=s1.search(20);
        System.out.println("Index of 20 : "+index); // 3
    }
}
```

```

        System.out.println();

        index=s1.search(99);
        System.out.println("Index of 99 : "+index); // -1
        System.out.println();

        int[] ar=s1.toArray();
        System.out.println(Arrays.toString(ar)); // [50,40,30,20,10]
    }
}

///////////////////////////////
Q116 From Q115
=====
import java.util.*;
class Stack{
    Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public boolean isEmpty(){
        return top==null;
    }
    public int size(){
        Node temp=top;
        int count=0;
        while(temp!=null){
            temp=temp.next;
            count++;
        }
        return count;
    }
    public void printStack(){
        System.out.print("[");
        Node temp=top;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println(isEmpty()? "empty]":"\b\b]");
    }
    public void pop(){
        top=top.next;
    }
    public void clear(){
        top=null;
    }
    public int peek(){

```

```

        return isEmpty() ? -1: top.data;
    }
    public int search(int data){
        if(top==null){
            return -1;
        }
        Node temp=top;
        int index=0;
        while(temp!=null){
            if(temp.data==data){
                return index;
            }
            index++;
            temp=temp.next;
        }
        return -1;
    }
    public int[] toArray(){
        int[] tempdataArray=new int[size()];
        Node temp=top;
        for (int i = 0; i < tempdataArray.length; i++){
            tempdataArray[i]=temp.data;
            temp=temp.next;
        }
        return tempdataArray;
    }
    public int poll(){
        if(isEmpty()){
            return -1;
        }
        int data=top.data;
        pop();
        return data;
    }
}
class Node{
    int data;
    Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
    }
}

```

```

        System.out.println();

        int index=s1.search(20);
        System.out.println("Index of 20 : "+index); //3
        System.out.println();

        index=s1.search(99);
        System.out.println("Index of 99 : "+index); //-1
        System.out.println();

        int[] ar=s1.toArray();
        System.out.println(Arrays.toString(ar)); //[50,40,30,20,10]
    }
}

```

//////////
Q117 From Q116

=====

```

public int search(int data){
    Node temp=top;
    int index=-1;
    while(temp!=null){
        index++;
        if(temp.data==data){
            return index;
        }
        temp=temp.next;
    }
    return index;
}

```

//////////
Q118 Step XIII

=====

```

import java.util.*;
class Stack{
    private Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
    public boolean isEmpty(){
        return top==null;
    }
}

```

```
public int size(){
    Node temp=top;
    int count=0;
    while(temp!=null){
        temp=temp.next;
        count++;
    }
    return count;
}
public void printStack(){
    System.out.print("[");
    Node temp=top;
    while(temp!=null){
        System.out.print(temp.data+", ");
        temp=temp.next;
    }
    System.out.println(isEmpty()? "empty]":"\b\b]");
}
public void pop(){
    top=top.next;
}
public void clear(){
    top=null;
}
public int peek(){
    return isEmpty() ? -1: top.data;
}
public int search(int data){
    Node temp=top;
    int index=-1;
    while(temp!=null){
        index++;
        if(temp.data==data){
            return index;
        }
        temp=temp.next;
    }
    return index;
}
public int[] toArray(){
    int[] tempdataArray=new int[size()];
    Node temp=top;
    for (int i = 0; i < tempdataArray.length; i++){
        tempdataArray[i]=temp.data;
        temp=temp.next;
    }
    return tempdataArray;
}
```

```

public int poll(){
    if(isEmpty()){
        return -1;
    }
    int data=top.data;
    pop();
    return data;
}
}

class Node{
    private int data;
    private Node next;
    Node(int data){
        this.data=data;
    }
}

class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println();

        int index=s1.search(20);
        System.out.println("Index of 20 : "+index); // 3
        System.out.println();

        index=s1.search(99);
        System.out.println("Index of 99 : "+index); // -1
        System.out.println();

        int[] ar=s1.toArray();
        System.out.println(Arrays.toString(ar)); // [50,40,30,20,10]
    }
}
////////////////////////////////////////////////////////////////

```

Q119 From q118 (Using Inner classes)

=====

```

import java.util.*;
class Stack{
    private Node top;

    public void push(int data){
        Node n1=new Node(data);
        n1.next=top;
        top=n1;
    }
}

```

```
public boolean isEmpty(){
    return top==null;
}
public int size(){
    Node temp=top;
    int count=0;
    while(temp!=null){
        temp=temp.next;
        count++;
    }
    return count;
}
public void printStack(){
    System.out.print("[");
    Node temp=top;
    while(temp!=null){
        System.out.print(temp.data+", ");
        temp=temp.next;
    }
    System.out.println(isEmpty() ? "empty]":"\b\b]");
}
public void pop(){
    top=top.next;
}
public void clear(){
    top=null;
}
public int peek(){
    return isEmpty() ? -1: top.data;
}
public int search(int data){
    Node temp=top;
    int index=-1;
    while(temp!=null){
        index++;
        if(temp.data==data){
            return index;
        }
        temp=temp.next;
    }
    return index;
}
public int[] toArray(){
    int[] tempdataArray=new int[size()];
    Node temp=top;
    for (int i = 0; i < tempdataArray.length; i++){
        tempdataArray[i]=temp.data;
        temp=temp.next;
    }
    return tempdataArray;
}
```

```
public int poll(){
    if(isEmpty()){
        return -1;
    }
    int data=top.data;
    pop();
    return data;
}
class Node{
    private int data;
    private Node next;
    Node(int data){
        this.data=data;
    }
}
class Demo{
    public static void main(String args[]){
        Stack s1=new Stack();
        s1.push(10);
        s1.push(20);
        s1.push(30);
        s1.push(40);
        s1.push(50);
        s1.printStack(); // [50,40,30,20,10]
        System.out.println();

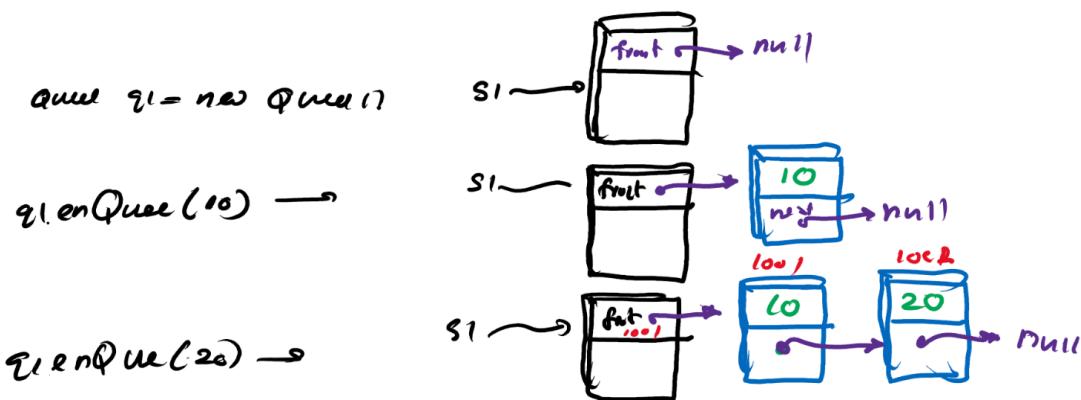
        int index=s1.search(20);
        System.out.println("Index of 20 : "+index); // 3
        System.out.println();

        index=s1.search(99);
        System.out.println("Index of 99 : "+index); // -1
        System.out.println();

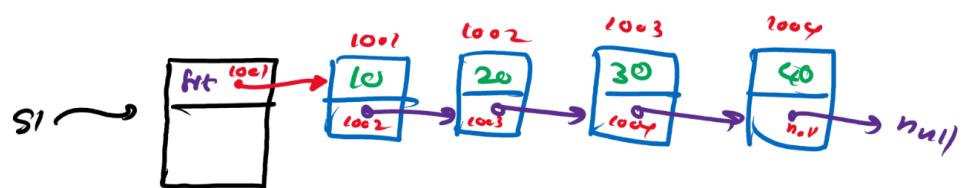
        int[] ar=s1.toArray();
        System.out.println(Arrays.toString(ar)); // [50,40,30,20,10]
    }
}
```

Class "Queue"

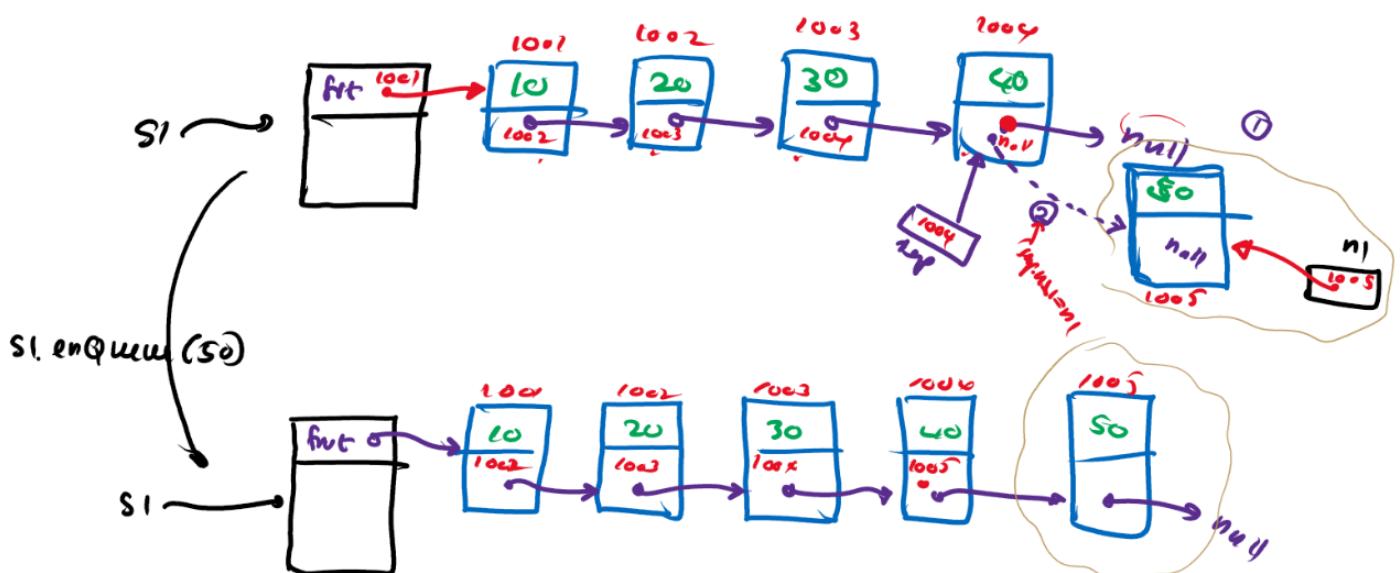
Q120



q1.enqueue(30)
q1.enqueue(40)



Q121



Q122

```
import java.util.*;  
class Queue{  
    private Node front;  
  
    public void enqueue(int data){  
        Node n1=new Node(data);  
        //  
        //  
    }  
}
```

```
public boolean isEmpty(){
    return front==null;
}
public int size(){
    Node temp=front;
    int count=0;
    while(temp!=null){
        temp=temp.next;
        count++;
    }
    return count;
}
public void printQueue(){
    System.out.print("[");
    Node temp=front;
    while(temp!=null){
        System.out.print(temp.data+", ");
        temp=temp.next;
    }
    System.out.println(isEmpty() ? "empty]":"\b\b]");
}
public void deQueue(){
    front=front.next;
}
public void clear(){
    front=null;
}
public int peek(){
    return isEmpty() ? -1: front.data;
}
public int search(int data){
    Node temp=front;
    int index=-1;
    while(temp!=null){
        index++;
        if(temp.data==data){
            return index;
        }
        temp=temp.next;
    }
    return index;
}
public int[] toArray(){
    int[] tempdataArray=new int[size()];
    Node temp=front;
    for (int i = 0; i < tempdataArray.length; i++){
        tempdataArray[i]=temp.data;
        temp=temp.next;
    }
    return tempdataArray;
}
```

```

public int poll(){
    if(isEmpty()){
        return -1;
    }
    int data=front.data;
    deQueue();
    return data;
}
class Node{
    private int data;
    private Node next;
    Node(int data){
        this.data=data;
    }
}
}

class Demo{
    public static void main(String args[]){
        Queue q1=new Queue();
        q1.enQueue(10);
        q1.enQueue(20);
        q1.enQueue(30);
        q1.enQueue(40);
        q1.enQueue(50);
        q1.printQueue();//[10,20,30,40,50]
    }
}

```

//////////////
Q123 From Q122

```

=====
import java.util.*;
class Queue{
    private Node front;

    public void enQueue(int data){
        Node n1=new Node(data);
        if(front==null){
            front=n1;
        }else{
            Node temp=front;
            while(temp.next!=null){
                temp=temp.next;
            }
            temp.next=n1;
        }
    }
    public boolean isEmpty(){
        return front==null;
    }
    public int size(){

```

```

        Node temp=front;
        int count=0;
        while(temp!=null){
            temp=temp.next;
            count++;
        }
        return count;
    }
    public void printQueue(){
        System.out.print("[");
        Node temp=front;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println(isEmpty()? "empty]":"\b\b]");
    }
    public void deQueue(){
        front=front.next;
    }
    public void clear(){
        front=null;
    }
    public int peek(){
        return isEmpty() ? -1: front.data;
    }
    public int search(int data){
        Node temp=front;
        int index=-1;
        while(temp!=null){
            index++;
            if(temp.data==data){
                return index;
            }
            temp=temp.next;
        }
        return index;
    }
    public int[] toArray(){
        int[] tempdataArray=new int[size()];
        Node temp=front;
        for (int i = 0; i < tempdataArray.length; i++){
            tempdataArray[i]=temp.data;
            temp=temp.next;
        }
        return tempdataArray;
    }
    public int poll(){
        if(isEmpty()){
            return -1;
        }

```

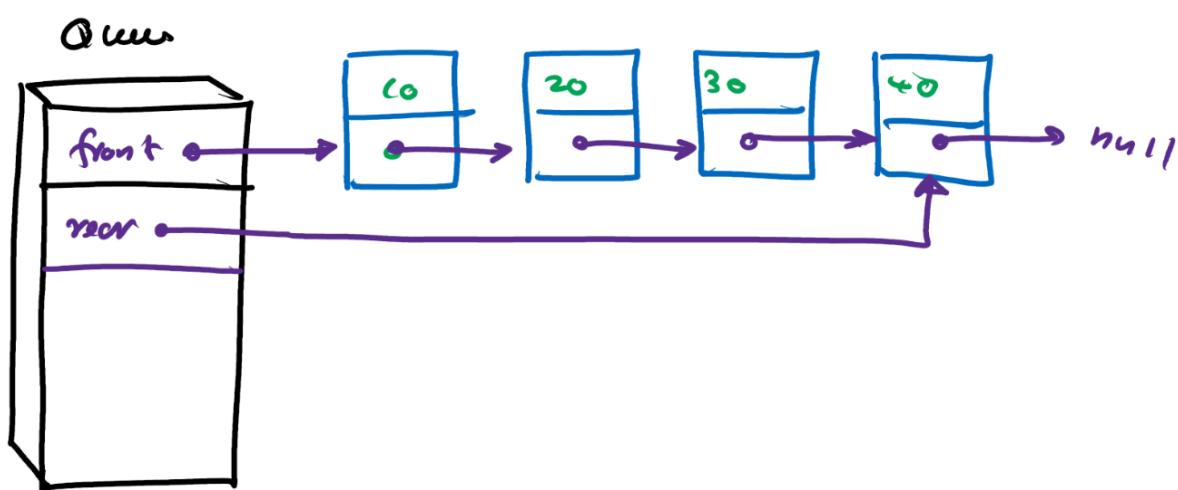
```

        int data=front.data;
        deQueue();
        return data;
    }
    class Node{
        private int data;
        private Node next;
        Node(int data){
            this.data=data;
        }
    }
}
class Demo{
    public static void main(String args[]){
        Queue q1=new Queue();
        q1.enQueue(10);
        q1.enQueue(20);
        q1.enQueue(30);
        q1.enQueue(40);
        q1.enQueue(50);
        q1.printQueue()//[10,20,30,40,50]

        q1.deQueue();
        q1.printQueue()//[20,30,40,50]
    }
}

```

//////////
Q124 Exercise
=====



```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Class “List”

```
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
```

Q125

```
import java.util.*;
class List{
    private Node start;

    public void add(int index, int data){
        //
    }

    public int get(int index){
        return -1;
    }

    public void remove(int index){
    }

    public void add(int data){
        Node n1=new Node(data);
        if(start==null){
            start=n1;
        }else{
            Node temp=start;
            while(temp.next!=null){
                temp=temp.next;
            }
            temp.next=n1;
        }
    }

    public boolean isEmpty(){
        return start==null;
    }

    public int size(){
        Node temp=start;
        int count=0;
        while(temp!=null){
            temp=temp.next;
            count++;
        }
        return count;
    }

    public void printList(){
        System.out.print("[");
        Node temp=start;
        while(temp!=null){
            System.out.print(temp.data+", ");
            temp=temp.next;
        }
        System.out.println(isEmpty()? "empty]":"\b\b]");
    }
}
```

```
public void clear(){
    start=null;
}

public int search(int data){
    Node temp=start;
    int index=-1;
    while(temp!=null){
        index++;
        if(temp.data==data){
            return index;
        }
        temp=temp.next;
    }
    return index;
}
public int[] toArray(){
    int[] tempdataArray=new int[size()];
    Node temp=start;
    for (int i = 0; i < tempdataArray.length; i++){
        tempdataArray[i]=temp.data;
        temp=temp.next;
    }
    return tempdataArray;
}
class Node{
    private int data;
    private Node next;
    Node(int data){
        this.data=data;
    }
}
}
class Demo{
    public static void main(String args[]){
        List list1=new List();
        list1.add(10);
        list1.add(20);
        list1.add(30);
        list1.add(40);
        list1.add(50);
        list1.printList(); // [10, 20, 30, 40, 50];

        list1.add(2,222);
        list1.printList(); // [10, 20, 222, 30, 40, 50];

        list1.add(0,9);
        list1.printList(); // [9, 10, 20, 222, 30, 40, 50];

        list1.add(list1.size()-1, 60);
        list1.printList(); // [9, 10, 20, 222, 30, 40, 50, 60];
    }
}
```

```

list1.remove(list1.size()-1);
list1.printList(); // [9,10,20,222,30,40,50];

list1.remove(0);
list1.printList(); // [10,20,222,30,40,50];

list1.remove(2);
list1.printList(); // [10,20,30,40,50];
}

}

```

//

Q126

```

import java.util.*;
class List{
    private Node start;

    public void add(int index, int data){
        Node n1=new Node(data);
        if(index>=0 && index<=size()){
            if(index==0){
                n1.next=start;
                start=n1;
            }else{
                int count=0;
                Node temp=start;
                while(count<index-1){
                    temp=temp.next;
                    count++;
                }
                n1.next=temp.next;
                temp.next=n1;
            }
        }
    }

    public int get(int index){
        if(index>=0 && index<size()){
            int count=0;
            Node temp=start;
            while(count<index){
                temp=temp.next;
                count++;
            }
            return temp.data;
        }
        return -1;
    }

    public void remove(int index){
        if(index>=0 && index<size()){
            if(index==0){

```

```

        start=start.next;
    }else{
        int count=0;
        Node temp=start;
        while(count<index-1){
            temp=temp.next;
            count++;
        }
        temp.next=temp.next.next;
    }
}

public void add(int data){
    Node n1=new Node(data);
    if(start==null){
        start=n1;
    }else{
        Node temp=start;
        while(temp.next!=null){
            temp=temp.next;
        }
        temp.next=n1;
    }
}

public boolean isEmpty(){
    return start==null;
}

public int size(){
    Node temp=start;
    int count=0;
    while(temp!=null){
        temp=temp.next;
        count++;
    }
    return count;
}

public void printList(){
    System.out.print("[");
    Node temp=start;
    while(temp!=null){
        System.out.print(temp.data+", ");
        temp=temp.next;
    }
    System.out.println(isEmpty()? "empty]":"\b\b]");
}

public void clear(){
    start=null;
}

public int search(int data){

```

```

Node temp=start;
int index=-1;
while(temp!=null){
    index++;
    if(temp.data==data){
        return index;
    }
    temp=temp.next;
}
return index;
}

public int[] toArray(){
    int[] tempdataArray=new int[size()];
    Node temp=start;
    for (int i = 0; i < tempdataArray.length; i++){
        tempdataArray[i]=temp.data;
        temp=temp.next;
    }
    return tempdataArray;
}

class Node{
    private int data;
    private Node next;
    Node(int data){
        this.data=data;
    }
}

}

class Demo{
    public static void main(String args[]){
        List list1=new List();
        list1.add(10);
        list1.add(20);
        list1.add(30);
        list1.add(40);
        list1.add(50);
        list1.printList(); // [10,20,30,40,50];

        list1.add(2,222);
        list1.printList(); // [10,20,222,30,40,50];

        list1.add(0,9);
        list1.printList(); // [9,10,20,222,30,40,50];

        list1.add(list1.size(), 60);
        list1.printList(); // [9,10,20,222,30,40,50,60];

        System.out.println();
        list1.remove(list1.size()-1);
        list1.printList(); // [9,10,20,222,30,40,50];
    }
}

```

```
list1.remove(0);
list1.printList() // [10,20,222,30,40,50];

list1.remove(2);
list1.printList() // [10,20,30,40,50];
}

} //////////////////////////////////////////////////////////////////
```

Java Swing Applications

```
////////////////////////////////////////////////////////////////
Q127
import javax.swing.*;
class Demo{
    public static void main(String args[]){
        JFrame f1=new JFrame();
        f1.setSize(300,300);
        f1.setTitle("Calculator");
        f1.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        f1.setLocationRelativeTo(null);
        f1.setVisible(true);

    }
}

////////////////////////////////////////////////////////////////
Q128
import javax.swing.*;
class Demo{
    public static void main(String args[]){
        Calculator c1=new Calculator();
        c1.setSize(300,300);
        c1.setTitle("Calculator");
        c1.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        c1.setLocationRelativeTo(null);
        c1.setVisible(true);

    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q129

```
import javax.swing.*;
class Calculator{

}

class Demo{
    public static void main(String args[]){
        Calculator c1=new Calculator();
        c1.setSize(300,300);
        c1.setTitle("Calculator");
        c1.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        c1.setLocationRelativeTo(null);
        c1.setVisible(true);

    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q130

```
import javax.swing.*;
class Calculator{
    public void setSize(int l, int w){

    }
    public void setTitle(String title){

    }
    public void setDefaultCloseOperation(int flag){

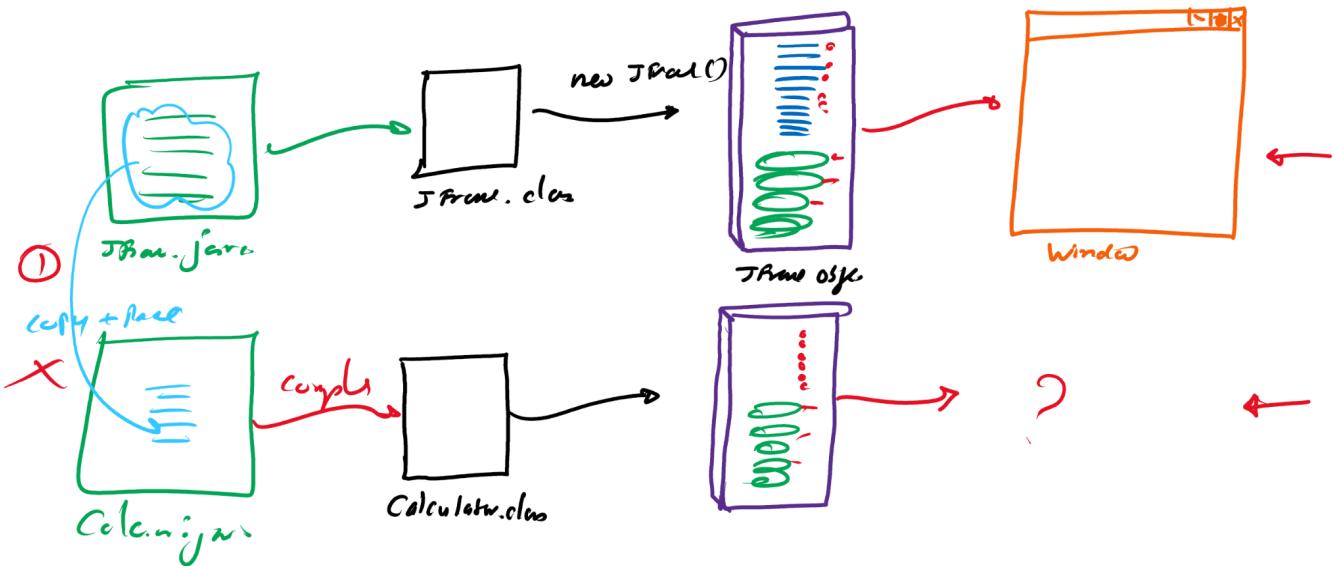
    }
    public void setLocationRelativeTo(Object obj){

    }
    public void setVisible(boolean flag){

    }
}

class Demo{
    public static void main(String args[]){
        Calculator c1=new Calculator();
        c1.setSize(300,300);
        c1.setTitle("Calculator");
        c1.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        c1.setLocationRelativeTo(null);
        c1.setVisible(true);

    }
}
```



.....

0131

```
import javax.swing.*;  
class Calculator extends JFrame{
```

}

```
class Demo{
```

```
public static void main(String args[]){
    Calculator c1=new Calculator();
    c1.setSize(300,300);
    c1.setTitle("Calculator");
    c1.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    c1.setLocationRelativeTo(null);
    c1.setVisible(true);
```

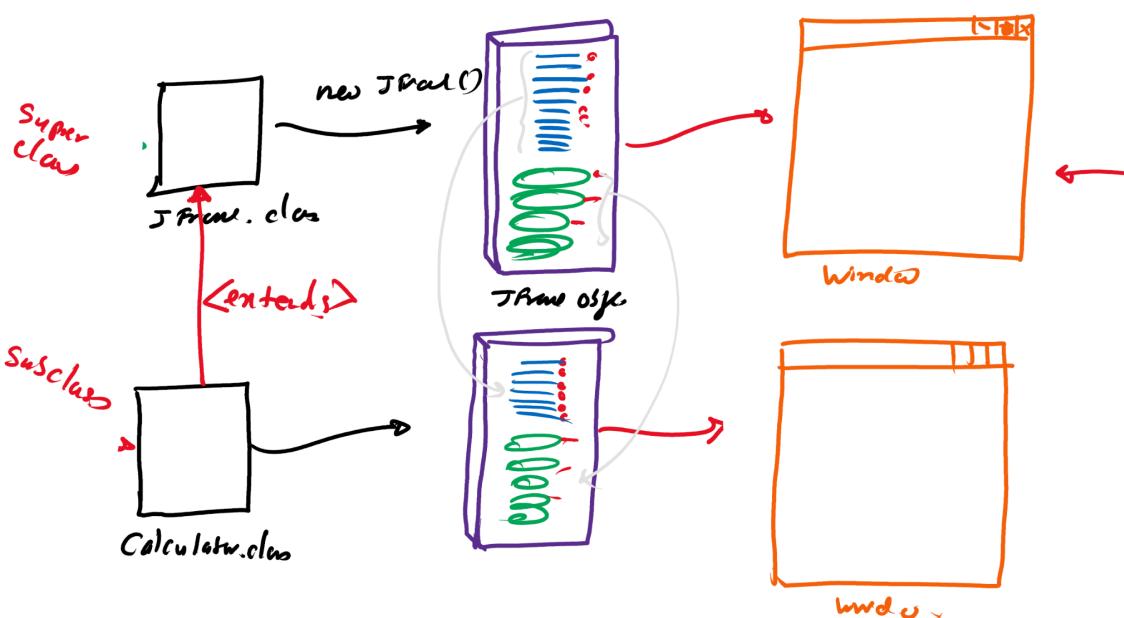
}

```
}
```

```
//Calculator IS A JFrame
```

```
//Calculator==>Subclass
```

```
//JFrame ==>Super class
```



||||||||||||||||||||||||||||||||||||||||||||||||||||

Q132

Java Swing application

- 1. Containers (JFrame, JWindow, JPanel, JDialog ----)
- 2. controller (JButton, JTextField, JLabel, ImageIcon, --)
- 3. Layout manager (BoxLayout, GridLayout, FlowLayout --)
- 4. Event Handlly (-----)

||||||||||||||||||||||||||||||||||||||||||||

Q133

```
import javax.swing.*;  
class Calculator extends JFrame{  
    JButton exitButton;  
  
    Calculator(){  
        setSize(300,300);  
        setTitle("Calculator");  
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);  
        setLocationRelativeTo(null);  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        new Calculator().setVisible(true);  
  
    }  
}
```

||||||||||||||||||||||||||||||||||||||||||||

Q134

```
import javax.swing.*;  
class Calculator extends JFrame{  
    JButton exitButton;  
  
    Calculator(){  
        setSize(300,300);  
        setTitle("Calculator");  
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);  
        setLocationRelativeTo(null);  
  
        exitButton=new JButton();  
        exitButton.setText("Exit");  
    }  
}
```

```
        add(exitButton);
    }
}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}

///////////////////////////////
Q135
import javax.swing.*;
class Calculator extends JFrame{
    JButton exitButton;
    JButton cancelButton;

    Calculator(){
        setSize(300,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        exitButton=new JButton();
        exitButton.setText("Exit");
        add(exitButton);

        cancelButton=new JButton("Cancel");
        add(cancelButton);
    }
}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}
```

```
///////////////////////////////
Q136 Layout Manager (BorderLayout)
=====

import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{
    JButton exitButton;
    JButton cancelButton;

    Calculator(){
        setSize(300,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
```

```

        setLocationRelativeTo(null);
        //Default layout->
        exitButton=new JButton();
        exitButton.setText("Exit");
        add("North",exitButton);

        cancelButton=new JButton("Cancel");
        add("South",cancelButton);
    }
}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}

```

//

Q137

```

import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{
    JButton southButton;
    JButton northButton;
    JButton centerButton;
    JButton westButton;
    JButton eastButton;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());

        southButton=new JButton("South");
        southButton.setFont(new Font("",1,25));
        add("South",southButton);

        northButton=new JButton("North");
        northButton.setFont(new Font("",1,25));
        add("North",northButton);

        westButton=new JButton("West");
        westButton.setFont(new Font("",1,25));
        add("West",westButton);

        eastButton=new JButton("East");
        eastButton.setFont(new Font("",1,25));
        add("East",eastButton);
    }
}

```

```

        centerButton=new JButton("Center");
        centerButton.setFont(new Font("",1,25));
        add("Center",centerButton);

    }

}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}

```

//

Q138 FlowLayout

=====

```

import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{
    JButton southButton;
    JButton northButton;
    JButton centerButton;
    JButton westButton;
    JButton eastButton;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new FlowLayout());

        southButton=new JButton("South");
        southButton.setFont(new Font("",1,25));
        add(southButton);

        northButton=new JButton("North");
        northButton.setFont(new Font("",1,25));
        add(northButton);

        westButton=new JButton("West");
        westButton.setFont(new Font("",1,25));
        add(westButton);

        eastButton=new JButton("East");
        eastButton.setFont(new Font("",1,25));
        add(eastButton);

        centerButton=new JButton("Center");
        centerButton.setFont(new Font("",1,25));
        add(centerButton);
    }
}

```

```

    }
}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}

///////////////////////////////
Q139
import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{
    JTextField textField;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new FlowLayout());

        textField=new JTextField(10);
        textField.setFont(new Font("",1,25));
        add(textField);
    }
}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}

```

```

///////////////////////////////
Q140 BorderLayout vs FlowLayout
=====

import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{
    JTextField textField;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());
    }
}
```

```
        textField=new JTextField(10);
        textField.setFont(new Font("",1,25));
        add("South",textField);
    }
}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}
```

//

Q141 GridLayout

=====

```
import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{
    JButton southButton;
    JButton northButton;
    JButton centerButton;
    JButton westButton;
    JButton eastButton;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        setLayout(new GridLayout(2,2));
        southButton=new JButton("South");
        southButton.setFont(new Font("",1,25));
        add(southButton);

        northButton=new JButton("North");
        northButton.setFont(new Font("",1,25));
        add(northButton);

        westButton=new JButton("West");
        westButton.setFont(new Font("",1,25));
        add(westButton);

        eastButton=new JButton("East");
        eastButton.setFont(new Font("",1,25));
        add(eastButton);

    }
}
```

```
class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);
    }
}

///////////////////////////////
Q142
import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{
    JButton bt0;
    JButton bt1;
    JButton bt2;
    JButton bt3;
    JButton bt4;
    JButton bt5;
    JButton bt6;
    JButton bt7;
    JButton bt8;
    JButton bt9;
    JButton btAdd;
    JButton btMul;
    JButton btDiv;
    JButton btSub;
    JButton btDot;
    JButton btEquals;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        setLayout(new GridLayout(4,4));

        bt7=new JButton("7");
        bt7.setFont(new Font("",1,20));
        add(bt7);

        bt8=new JButton("8");
        bt8.setFont(new Font("",1,20));
        add(bt8);

        bt9=new JButton("9");
        bt9.setFont(new Font("",1,20));
        add(bt9);

        btMul=new JButton("*");
        btMul.setFont(new Font("",1,20));
    }
}
```

```
add(btMul);

bt4=new JButton("4");
bt4.setFont(new Font("",1,20));
add(bt4);

bt5=new JButton("5");
bt5.setFont(new Font("",1,20));
add(bt5);

bt6=new JButton("6");
bt6.setFont(new Font("",1,20));
add(bt6);

btDiv=new JButton("/");
btDiv.setFont(new Font("",1,20));
add(btDiv);

bt1=new JButton("1");
bt1.setFont(new Font("",1,20));
add(bt1);

bt2=new JButton("2");
bt2.setFont(new Font("",1,20));
add(bt2);

bt3=new JButton("3");
bt3.setFont(new Font("",1,20));
add(bt3);

btAdd=new JButton("+");
btAdd.setFont(new Font("",1,20));
add(btAdd);

bt0=new JButton("0");
bt0.setFont(new Font("",1,20));
add(bt0);

btDot=new JButton(".");
btDot.setFont(new Font("",1,20));
add(btDot);

btEquals=new JButton "=";
btEquals.setFont(new Font("",1,20));
add(btEquals);

btSub=new JButton "-";
btSub.setFont(new Font("",1,20));
add(btSub);
```

```

        pack();

    }

}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}

///////////////////////////////
Q143 From Q142
=====
import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{

    JButton[] buttonArray;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        setLayout(new GridLayout(4,4));

        String[] buttonNameArray={"7","8","9","*","4","5","6","/","1","2","3","+","0",".","=","-"};
        buttonArray=new JButton[16];
        for(int i=0; i<16; i++){
            buttonArray[i]=new JButton(buttonNameArray[i]);
            buttonArray[i].setFont(new Font("",1,20));
            add(buttonArray[i]);
        }

        pack();
    }
}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);

    }
}

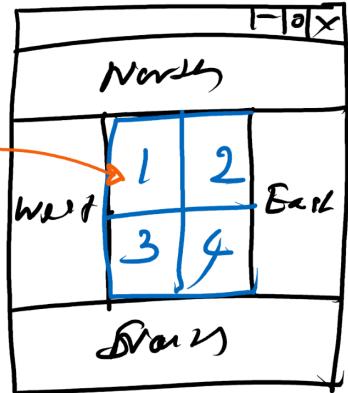
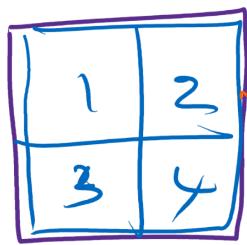
```

||||||||||||||||||||||||||||||||||||||||||||||||

Q144 Class "JPanel"

=====

JPanel p1=new JPanel();
p1.setLayout(new GridLayout(2,2));



||||||||||||||||||||||||||||||||||||||||||||

Q145 From Q144

=====

```
import javax.swing.*;  
import java.awt.*;  
class Calculator extends JFrame{  
    JButton southButton;  
    JButton northButton;  
    JButton westButton;  
    JButton eastButton;  
  
    JButton bt1;  
    JButton bt2;  
    JButton bt3;  
    JButton bt4;  
  
    Calculator(){  
        setSize(400,300);  
        setTitle("Calculator");  
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);  
        setLocationRelativeTo(null);  
        setLayout(new BorderLayout());  
  
        southButton=new JButton("South");  
        southButton.setFont(new Font("",1,25));  
        add("South",southButton);  
  
        northButton=new JButton("North");  
        northButton.setFont(new Font("",1,25));  
        add("North",northButton);  
  
        westButton=new JButton("West");  
        westButton.setFont(new Font("",1,25));  
        add("West",westButton);  
    }  
}
```

```

eastButton=new JButton("East");
eastButton.setFont(new Font("",1,25));
add("East",eastButton);

JPanel centerPanel=new JPanel();
centerPanel.setLayout(new GridLayout(2,2));

bt1=new JButton("1");
bt1.setFont(new Font("",1,20));
centerPanel.add(bt1);
bt2=new JButton("2");
bt2.setFont(new Font("",1,20));
centerPanel.add(bt2);
bt3=new JButton("3");
bt3.setFont(new Font("",1,20));
centerPanel.add(bt3);
bt4=new JButton("4");
bt4.setFont(new Font("",1,20));
centerPanel.add(bt4);

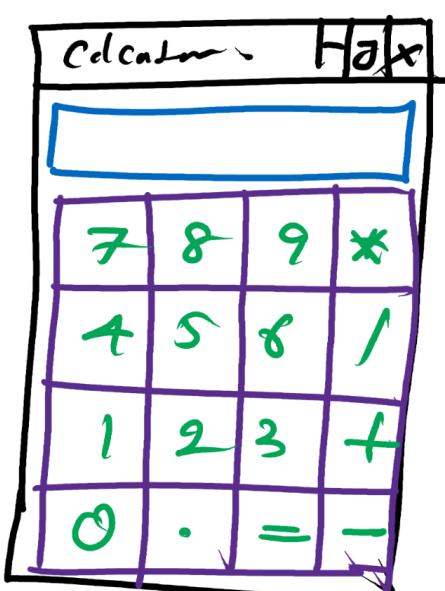
add("Center",centerPanel);
}

}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);
    }
}

```

//////////
Q146 Exercise
=====



```
////////////////////////////////////////////////////////////////////////
Q147 From Q146
=====
import javax.swing.*;
import java.awt.*;
class Calculator extends JFrame{

    JButton[] buttonArray;
    JTextField txtDisplay;

    Calculator(){
        setSize(400,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new BorderLayout());

        txtDisplay=new JTextField();
        txtDisplay.setFont(new Font("",1,20));
        add("North",txtDisplay);

        String[] buttonNameArray={"7","8","9","*","4","5","6","/","1","2","3","+","0",".","=","-"};
        buttonArray=new JButton[16];
        JPanel buttonPanel=new JPanel(new GridLayout(4,4,2,2));

        for(int i=0; i<16; i++){
            buttonArray[i]=new JButton(buttonNameArray[i]);
            buttonArray[i].setFont(new Font("",1,20));
            buttonPanel.add(buttonArray[i]);
        }
        add("Center",buttonPanel);
        pack();
    }

}

class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);
    }
}
```

```
Q149 Step II
=====
import javax.swing.*;
import java.awt.*;
class AddCustomerForm extends JFrame{
    private JLabel titleLabel;
    AddCustomerForm(){
        setSize(500,300);
        setTitle("Add Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        titleLabel=new JLabel("Add Customer Form");
        titleLabel.setHorizontalAlignment(JLabel.CENTER);
        titleLabel.setFont(new Font("",1,30));
        add("North",titleLabel);
    }
}
class Demo{
    public static void main(String args[]){
        new AddCustomerForm().setVisible(true);
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q150 Step III

=====

```
import javax.swing.*;
import java.awt.*;
class AddCustomerForm extends JFrame{
    private JLabel titleLabel;
    private JButton btnAdd;
    private JButton btnCancel;
    AddCustomerForm(){
        setSize(500,300);
        setTitle("Add Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        titleLabel=new JLabel("Add Customer Form");
        titleLabel.setHorizontalAlignment(JLabel.CENTER);
        titleLabel.setFont(new Font("",1,30));
        add("North",titleLabel);
        JPanel buttonPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));
        btnAdd=new JButton("Add");
        btnAdd.setFont(new Font("",1,20));
        buttonPanel.add(btnAdd);
        btnCancel=new JButton("Cancel");
        btnCancel.setFont(new Font("",1,20));
        buttonPanel.add(btnCancel);
        add("South",buttonPanel);
    }
}
class Demo{
    public static void main(String args[]){
        new AddCustomerForm().setVisible(true);
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q151 Step IV

=====

```
import javax.swing.*;
import java.awt.*;
class AddCustomerForm extends JFrame{
    private JLabel titleLabel;
    private JButton btnAdd;
    private JButton btnCancel;
    private JLabel lblId;
    private JLabel lblName;
    private JLabel lblAddress;
    private JLabel lblSalary;

    AddCustomerForm(){
        setSize(500,300);
        setTitle("Add Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        titleLabel=new JLabel("Add Customer Form");
        titleLabel.setHorizontalAlignment(JLabel.CENTER);
        titleLabel.setFont(new Font("",1,30));
        add("North",titleLabel);

        JPanel buttonPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));
        btnAdd=new JButton("Add");
        btnAdd.setFont(new Font("",1,20));
        buttonPanel.add(btnAdd);
        btnCancel=new JButton("Cancel");
        btnCancel.setFont(new Font("",1,20));
        buttonPanel.add(btnCancel);
        add("South",buttonPanel);

        JPanel labelPanel=new JPanel(new GridLayout(4,1));

        lblId=new JLabel("Id");
        lblId.setFont(new Font("",1,20));
        labelPanel.add(lblId);

        lblName=new JLabel("Name");
        lblName.setFont(new Font("",1,20));
        labelPanel.add(lblName);

        lblAddress=new JLabel("Address");
        lblAddress.setFont(new Font("",1,20));
        labelPanel.add(lblAddress);
```

```

lblSalary=new JLabel("Salary");
lblSalary.setFont(new Font("",1,20));
labelPanel.add(lblSalary);

add("West",labelPanel);

}

}

class Demo{
    public static void main(String args[]){
        new AddCustomerForm().setVisible(true);
    }
}

```

//////////
Q152 Step V

=====

```

import javax.swing.*;
import java.awt.*;

```

```
class AddCustomerForm extends JFrame{
```

```
    private JLabel titleLabel;
```

```
    private JButton btnAdd;
```

```
    private JButton btnCancel;
```

```
    private JLabel lblId;
```

```
    private JLabel lblName;
```

```
    private JLabel lblAddress;
```

```
    private JLabel lblSalary;
```

```
    private JTextField txtId;
```

```
    private JTextField txtName;
```

```
    private JTextField txtAddress;
```

```
    private JTextField txtSalary;
```

```
AddCustomerForm(){
```

```
    setSize(500,300);
```

```
    setTitle("Add Customer Form");
```

```
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
```

```
    setLocationRelativeTo(null);
```

```
    titleLabel=new JLabel("Add Customer Form");
```

```
    titleLabel.setHorizontalTextPosition(JLabel.CENTER);
```

```
    titleLabel.setFont(new Font("",1,30));
```

```
    add("North",titleLabel);
```

```
JPanel buttonPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));
```

```
    btnAdd=new JButton("Add");
```

```
    btnAdd.setFont(new Font("",1,20));
```

```
    buttonPanel.add(btnAdd);
```

```
btnCancel=new JButton("Cancel");
btnCancel.setFont(new Font("",1,20));
buttonPanel.add(btnCancel);
add("South",buttonPanel);

JPanel labelPanel=new JPanel(new GridLayout(4,1));

lblId=new JLabel("Id");
lblId.setFont(new Font("",1,20));
labelPanel.add(lblId);

lblName=new JLabel("Name");
lblName.setFont(new Font("",1,20));
labelPanel.add(lblName);

lblAddress=new JLabel("Address");
lblAddress.setFont(new Font("",1,20));
labelPanel.add(lblAddress);

lblSalary=new JLabel("Salary");
lblSalary.setFont(new Font("",1,20));
labelPanel.add(lblSalary);

add("West",labelPanel);

JPanel textPanel=new JPanel(new GridLayout(4,1));
txtId=new JTextField(4);
txtId.setFont(new Font("",1,20));
textPanel.add(txtId);

txtName=new JTextField(10);
txtName.setFont(new Font("",1,20));
textPanel.add(txtName);

txtAddress=new JTextField(25);
txtAddress.setFont(new Font("",1,20));
textPanel.add(txtAddress);

txtSalary=new JTextField(6);
txtSalary.setFont(new Font("",1,20));
textPanel.add(txtSalary);

add("East",textPanel);
}

}

class Demo{
    public static void main(String args[]){
        new AddCustomerForm().setVisible(true);
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q153 Step VI

=====

```
import javax.swing.*;
import java.awt.*;
class AddCustomerForm extends JFrame{
    private JLabel titleLabel;
    private JButton btnAdd;
    private JButton btnCancel;
    private JLabel lblId;
    private JLabel lblName;
    private JLabel lblAddress;
    private JLabel lblSalary;
    private JTextField txtId;
    private JTextField txtName;
    private JTextField txtAddress;
    private JTextField txtSalary;

    AddCustomerForm(){
        setSize(500,300);
        setTitle("Add Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        titleLabel=new JLabel("Add Customer Form");
        titleLabel.setHorizontalAlignment(JLabel.CENTER);
        titleLabel.setFont(new Font("",1,30));
        add("North",titleLabel);

        JPanel buttonPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));
        btnAdd=new JButton("Add");
        btnAdd.setFont(new Font("",1,20));
        buttonPanel.add(btnAdd);
        btnCancel=new JButton("Cancel");
        btnCancel.setFont(new Font("",1,20));
        buttonPanel.add(btnCancel);
        add("South",buttonPanel);

        JPanel labelPanel=new JPanel(new GridLayout(4,1));

        lblId=new JLabel("Id");
        lblId.setFont(new Font("",1,20));
        labelPanel.add(lblId);

        lblName=new JLabel("Name");
        lblName.setFont(new Font("",1,20));
        labelPanel.add(lblName);
```

```

lblAddress=new JLabel("Address");
lblAddress.setFont(new Font("",1,20));
labelPanel.add(lblAddress);

lblSalary=new JLabel("Salary");
lblSalary.setFont(new Font("",1,20));
labelPanel.add(lblSalary);

add("West",labelPanel);

JPanel textPanel=new JPanel(new GridLayout(4,1));
txtId=new JTextField(4);
txtId.setFont(new Font("",1,20));
JPanel idTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
idTextPanel.add(txtId);
textPanel.add(idTextPanel);

txtName=new JTextField(15);
txtName.setFont(new Font("",1,20));
JPanel nameTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
nameTextPanel.add(txtName);
textPanel.add(nameTextPanel);

txtAddress=new JTextField(20);
txtAddress.setFont(new Font("",1,20));
JPanel addressTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
addressTextPanel.add(txtAddress);
textPanel.add(addressTextPanel);

txtSalary=new JTextField(6);
txtSalary.setFont(new Font("",1,20));
JPanel salaryTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
salaryTextPanel.add(txtSalary);
textPanel.add(salaryTextPanel);

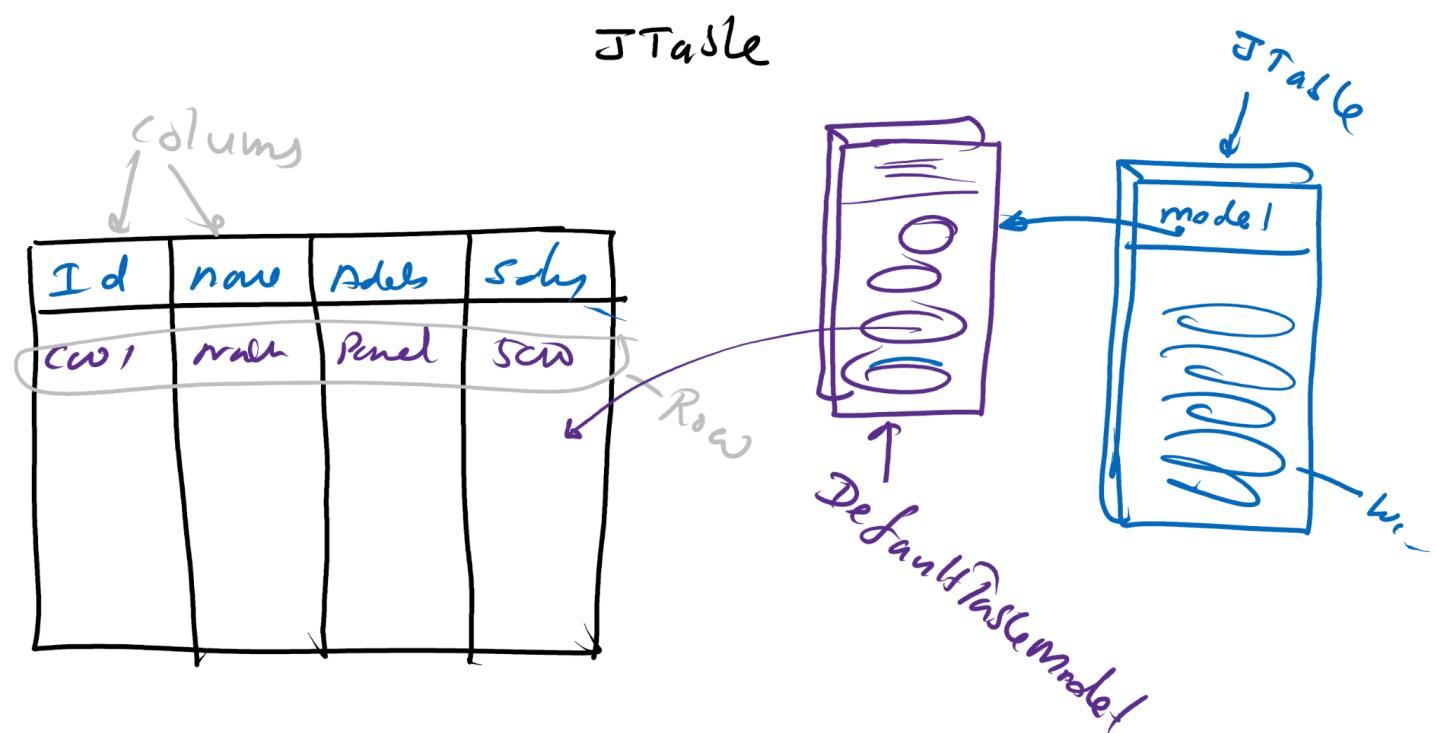
add("Center",textPanel);
}

}

class Demo{
    public static void main(String args[]){
        new AddCustomerForm().setVisible(true);
    }
}

```

Working with a JTable



Q154 Step I

```
=====
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
class ViewCustomerForm extends JFrame{
    private JTable tblCustomerDetails;
    private DefaultTableModel dtm;

    ViewCustomerForm(){
        setSize(400,300);
        setTitle("View Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
    }
}
class Demo{
    public static void main(String args[]){
        new ViewCustomerForm().setVisible(true);
    }
}
```

//////////

Q155 Step II

=====

```
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
class ViewCustomerForm extends JFrame{
    private JTable tblCustomerDetails;
    private DefaultTableModel dtm;

    ViewCustomerForm(){
        setSize(400,300);
        setTitle("View Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        String[] columnsName={"Customer Id","Name","Address","Salary"};
        dtm=new DefaultTableModel(columnsName,0);

        tblCustomerDetails=new JTable(dtm);

        add("Center",tblCustomerDetails);

    }
}

class Demo{
    public static void main(String args[]){
        new ViewCustomerForm().setVisible(true);
    }
}
```

//////////

Q156 Step IV

=====

```
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
class ViewCustomerForm extends JFrame{
    private JTable tblCustomerDetails;
    private DefaultTableModel dtm;

    private JLabel titleLabel;

    private JButton btnReload;
    ViewCustomerForm(){
        setSize(400,300);
        setTitle("View Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        titleLabel=new JLabel("View Customer Form");

    }
}
```

```

titleLabel.setHorizontalAlignment(JLabel.CENTER);
titleLabel.setFont(new Font("",1,30));
add("North",titleLabel);

String[] columnsName={"Customer Id","Name","Address","Salary"};
dtm=new DefaultTableModel(columnsName,0);

tblCustomerDetails=new JTable(dtm);

JScrollPane tablePane=new JScrollPane(tblCustomerDetails);

add("Center",tablePane);

JPanel buttonPanel=new JPanel(); //Default layout ->JPanel --FlowLayout
btnReload=new JButton("Reload");
btnReload.setFont(new Font("",1,20));
buttonPanel.add(btnReload);
add("South",buttonPanel);

}

}

class Demo{
    public static void main(String args[]){
        new ViewCustomerForm().setVisible(true);
    }
}

```

//
Event Handling

Q157 ActionEvent vs ActionListener

```

=====
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class Calculator extends JFrame{
    JButton exitButton;

    Calculator(){
        setSize(300,300);
        setTitle("Calculator");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);
        setLayout(new FlowLayout());
        exitButton=new JButton();
        exitButton.setText("Exit");

        exitButton.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                System.out.println("Exited..");
                System.exit(0);
            }
        });
    }
}
```

```

        }
    });

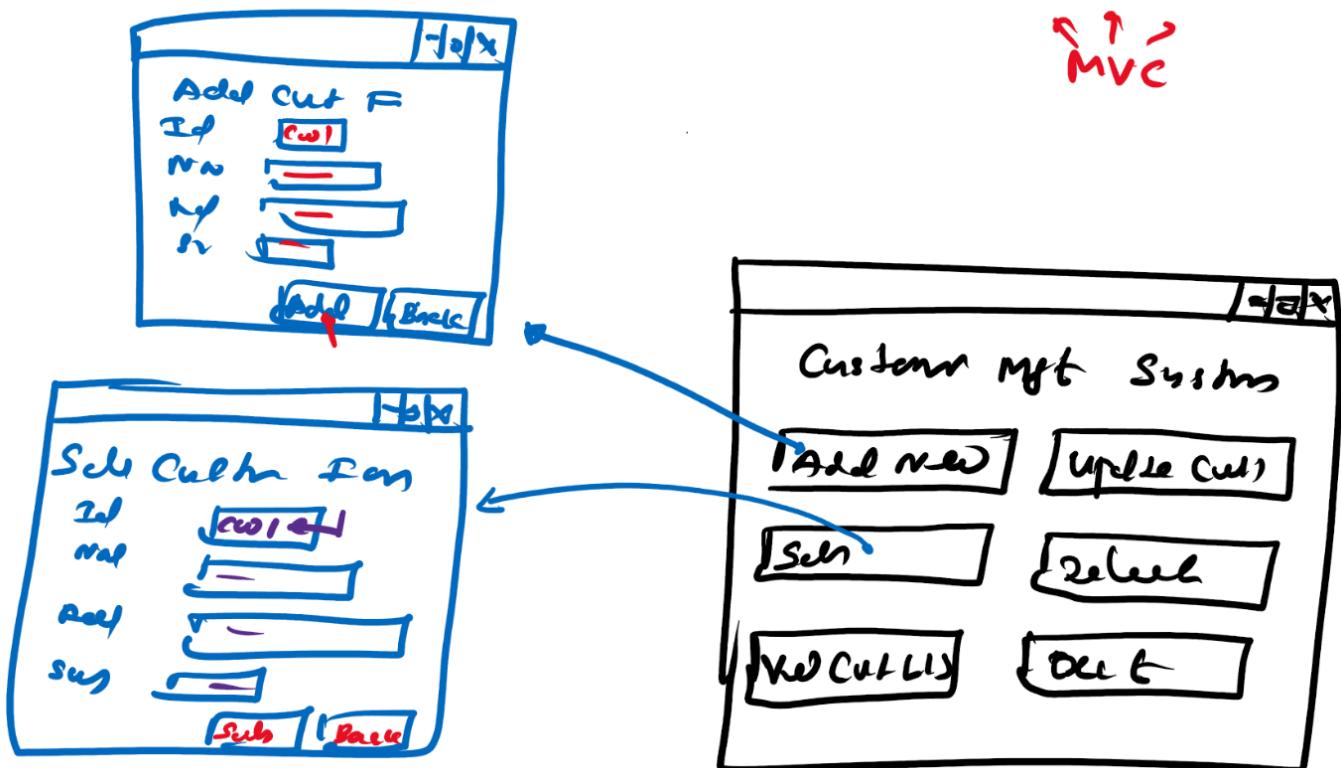
    add(exitButton);
}

}

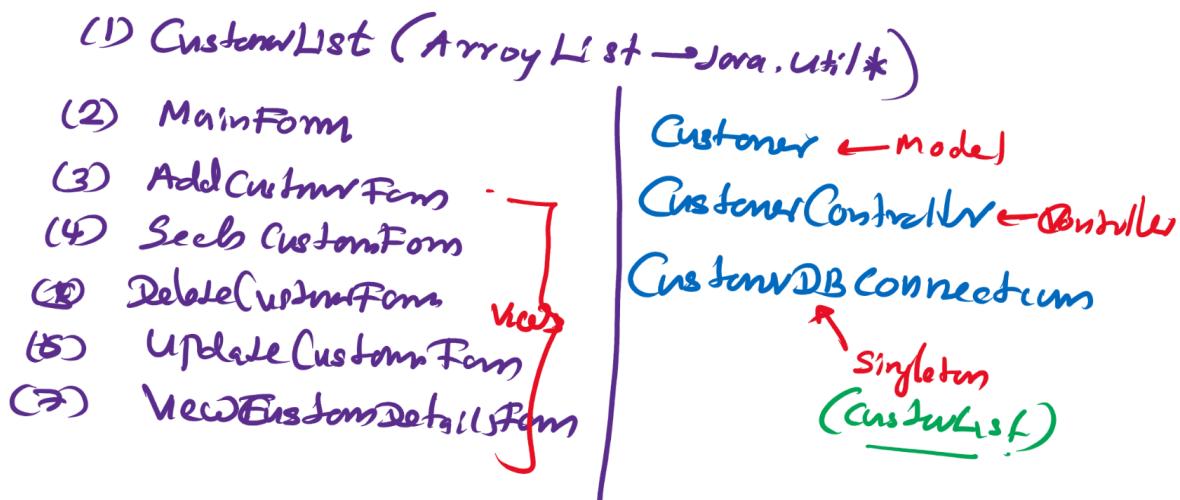
class Demo{
    public static void main(String args[]){
        new Calculator().setVisible(true);
    }
}

```

||||||||||||||||||||||||||||||||||||||||||||||
Q158



||||||||||||||||||||||||||||||||||||||
Q159



```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Q160

```
//-----Customer.java-----
```

```
class Customer{
```

```
    private String id;  
    private String name;  
    private String address;  
    private double salary;
```

```
    public Customer(){
```

```
}
```

```
    public Customer(String id, String name, String address, double salary){
```

```
        this.id=id;  
        this.name=name;  
        this.address=address;  
        this.salary=salary;
```

```
}
```

```
    public void setId(String id){
```

```
        this.id=id;
```

```
}
```

```
    public String getId(){
```

```
        return id;
```

```
}
```

```
    public String getName(){
```

```
        return name;
```

```
}
```

```
    public String getAddress(){
```

```
        return address;
```

```
}
```

```
    public double getSalary(){
```

```
        return salary;
```

```
}
```

```
}
```

```
//-----CustomerController.java-----
```

```
//-----AddCustomer.java-----
```

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
class AddCustomerForm extends JFrame{
```

```
    private JLabel titleLabel;
```

```
    private JButton btnAdd;
```

```
    private JButton btnCancel;
```

```
    private JLabel lblId;
```

```
    private JLabel lblName;
```

```

private JLabel lblAddress;
private JLabel lblSalary;

private JTextField txtId;
private JTextField txtName;
private JTextField txtAddress;
private JTextField txtSalary;

AddCustomerForm(){
    setSize(500,300);
    setTitle("Add Customer Form");
    setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    setLocationRelativeTo(null);

    titleLabel=new JLabel("Add Customer Form");
    titleLabel.setHorizontalAlignment(JLabel.CENTER);
    titleLabel.setFont(new Font("",1,30));
    add("North",titleLabel);

    JPanel buttonPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));
    btnAdd=new JButton("Add");
    btnAdd.setFont(new Font("",1,20));
    btnAdd.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent evt){
            String id=txtId.getText();
            String name=txtName.getText();
            String address=txtAddress.getText();
            double salary=Double.parseDouble(txtSalary.getText());
            Customer customer=new Customer(id,name,address,salary);
            CustomerMainForm.customerList.add(customer);
        }
    });
    buttonPanel.add(btnAdd);

    btnCancel=new JButton("Cancel");
    btnCancel.setFont(new Font("",1,20));
    btnCancel.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent evt){
            AddCustomerForm.this.dispose();
        }
    });
    buttonPanel.add(btnCancel);

    add("South",buttonPanel);

    JPanel labelPanel=new JPanel(new GridLayout(4,1));

    lblId=new JLabel("Id");
    lblId.setFont(new Font("",1,20));
    labelPanel.add(lblId);

```

```
lblName=new JLabel("Name");
lblName.setFont(new Font("",1,20));
labelPanel.add(lblName);

lblAddress=new JLabel("Address");
lblAddress.setFont(new Font("",1,20));
labelPanel.add(lblAddress);

lblSalary=new JLabel("Salary");
lblSalary.setFont(new Font("",1,20));
labelPanel.add(lblSalary);

add("West",labelPanel);

JPanel textPanel=new JPanel(new GridLayout(4,1));
txtId=new JTextField(4);
txtId.setFont(new Font("",1,20));
JPanel idTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
idTextPanel.add(txtId);
textPanel.add(idTextPanel);

txtName=new JTextField(15);
txtName.setFont(new Font("",1,20));
JPanel nameTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
nameTextPanel.add(txtName);
textPanel.add(nameTextPanel);

txtAddress=new JTextField(20);
txtAddress.setFont(new Font("",1,20));
JPanel addressTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
addressTextPanel.add(txtAddress);
textPanel.add(addressTextPanel);

txtSalary=new JTextField(6);
txtSalary.setFont(new Font("",1,20));
JPanel salaryTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
salaryTextPanel.add(txtSalary);
textPanel.add(salaryTextPanel);

add("Center",textPanel);
}
```

```
//-----ViewCustomer.java-----
import javax.swing.*;
import javax.swing.table.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
class ViewCustomerForm extends JFrame{
    private JTable tblCustomerDetails;
    private DefaultTableModel dtm;

    private JLabel titleLabel;
    private JButton btnReload;
    ViewCustomerForm(){
        setSize(400,300);
        setTitle("View Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        titleLabel=new JLabel("View Customer Form");
        titleLabel.setHorizontalAlignment(JLabel.CENTER);
        titleLabel.setFont(new Font("",1,30));
        add("North",titleLabel);

        String[] columnsName={"Customer Id","Name","Address","Salary"};
        dtm=new DefaultTableModel(columnsName,0);

        tblCustomerDetails=new JTable(dtm);

        JScrollPane tablePane=new JScrollPane(tblCustomerDetails);

        add("Center",tablePane);

        JPanel buttonPanel=new JPanel(); //Default layout ->JPanel --FlowLayout
        btnReload=new JButton("Reload");
        btnReload.setFont(new Font("",1,20));
        btnReload.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                for(int i=0; i<CustomerMainForm.customerList.size(); i++){
                    Customer customer=CustomerMainForm.customerList.get(i);
                    Object[] rowData={customer.getId(),
customer.getName(),customer.getAddress(),customer.getSalary()};
                    dtm.addRow(rowData);
                }
            }
        });
        buttonPanel.add(btnReload);
        add("South",buttonPanel);

    }
}
```

```
//-----CustomerMainForm.java-----  
import javax.swing.*;  
import java.awt.*;  
import java.util.*;  
import java.awt.event.*;  
class CustomerMainForm extends JFrame{  
  
    public static ArrayList <Customer>customerList=new ArrayList<>();  
  
    private AddCustomerForm addCustomerForm;  
    private ViewCustomerForm viewCustomerForm;  
  
    private JButton btnAddCustomer;  
    private JButton btnSearchCustomer;  
    private JButton btnDeleteCustomer;  
    private JButton btnUpdateCustomer;  
    private JButton btnViewCustomer;  
    private JButton btnExit;  
  
    CustomerMainForm(){  
        setTitle("Customer mgt System");  
        setSize(500,300);  
        setDefaultCloseOperation(DO_NOTHING_ON_CLOSE);  
        setLocationRelativeTo(null);  
  
        JPanel buttonPanel=new JPanel(new GridLayout(3,2));  
  
        btnAddCustomer=new JButton("Add Customer");  
        btnAddCustomer.setFont(new Font("",1,25));  
        btnAddCustomer.addActionListener(new ActionListener(){  
            public void actionPerformed(ActionEvent evt){  
                if(addCustomerForm==null){  
                    addCustomerForm=new AddCustomerForm();  
                }  
                addCustomerForm.setVisible(true);  
            }  
        });  
        buttonPanel.add(btnAddCustomer);  
  
        btnUpdateCustomer=new JButton("Update Customer");  
        btnUpdateCustomer.setFont(new Font("",1,25));  
        buttonPanel.add(btnUpdateCustomer);  
  
        btnViewCustomer=new JButton("View Customer");  
        btnViewCustomer.setFont(new Font("",1,25));  
        buttonPanel.add(btnViewCustomer);  
        btnViewCustomer.addActionListener(new ActionListener(){  
            public void actionPerformed(ActionEvent evt){  
                if(viewCustomerForm==null){  
                    viewCustomerForm=new ViewCustomerForm();  
                }  
            }  
        });  
    }  
}
```

```

        }
        viewCustomerForm.setVisible(true);
    }

    add("Center",buttonPanel);

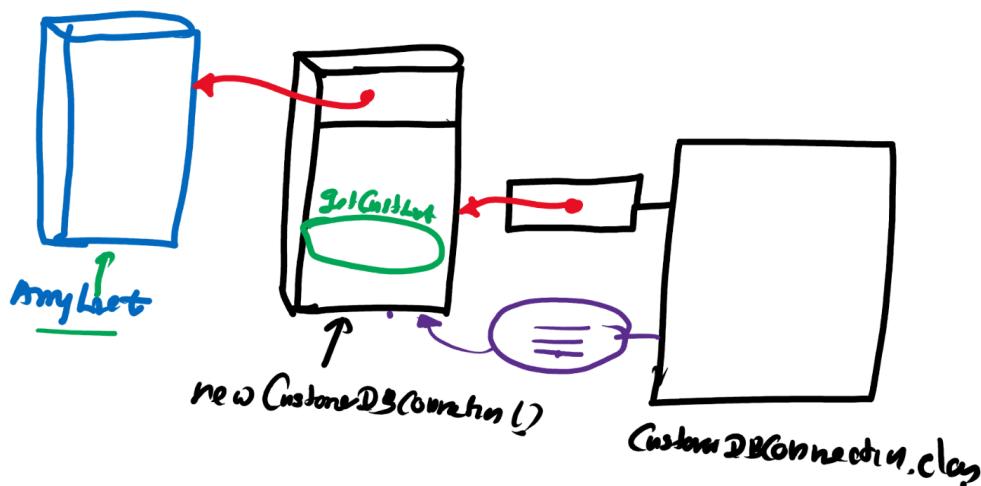
}

public static void main(String args[]){
    new CustomerMainForm().setVisible(true);
}
}

```

//////////

Design pattern “Singleton”



//////////

Q161

```

import java.util.*;
class CustomerDBConnection{
    private ArrayList <Customer>customerList;

    private static CustomerDBConnection customerDBConnection;
    private CustomerDBConnection(){
        customerList=new ArrayList<>();
    }
    public static CustomerDBConnection getInstance(){
        if(customerDBConnection==null){
            customerDBConnection=new CustomerDBConnection();
        }
        return customerDBConnection;
    }
    public ArrayList <Customer>getCustomerList(){
        return customerList;
    }
}

```

```
import javax.swing.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
class AddCustomerForm extends JFrame{
    private JLabel titleLabel;
    private JButton btnAdd;
    private JButton btnCancel;

    private JLabel lblId;
    private JLabel lblName;
    private JLabel lblAddress;
    private JLabel lblSalary;

    private JTextField txtId;
    private JTextField txtName;
    private JTextField txtAddress;
    private JTextField txtSalary;

    AddCustomerForm(){
        setSize(500,300);
        setTitle("Add Customer Form");
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(null);

        titleLabel=new JLabel("Add Customer Form");
        titleLabel.setHorizontalAlignment(JLabel.CENTER);
        titleLabel.setFont(new Font("",1,30));
        add("North",titleLabel);

        JPanel buttonPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));
        btnAdd=new JButton("Add");
        btnAdd.setFont(new Font("",1,20));
        btnAdd.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                String id=txtId.getText();
                String name=txtName.getText();
                String address=txtAddress.getText();
                double salary=Double.parseDouble(txtSalary.getText());
                Customer customer=new Customer(id,name,address,salary);
                CustomerDBConnection.getInstance().getCustomerList().add(customer);
            }
        });
        buttonPanel.add(btnAdd);
        btnCancel=new JButton("Cancel");
        btnCancel.setFont(new Font("",1,20));
        btnCancel.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent evt){
                AddCustomerForm.this.dispose();
            }
        });
    }
}
```

```
buttonPanel.add(btnCancel);
add("South",buttonPanel);
JPanel labelPanel=new JPanel(new GridLayout(4,1));

lblId=new JLabel("Id");
lblId.setFont(new Font("",1,20));
labelPanel.add(lblId);

lblName=new JLabel("Name");
lblName.setFont(new Font("",1,20));
labelPanel.add(lblName);

lblAddress=new JLabel("Address");
lblAddress.setFont(new Font("",1,20));
labelPanel.add(lblAddress);

lblSalary=new JLabel("Salary");
lblSalary.setFont(new Font("",1,20));
labelPanel.add(lblSalary);

add("West",labelPanel);

JPanel textPanel=new JPanel(new GridLayout(4,1));
txtId=new JTextField(4);
txtId.setFont(new Font("",1,20));
JPanel idTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
idTextPanel.add(txtId);
textPanel.add(idTextPanel);

txtName=new JTextField(15);
txtName.setFont(new Font("",1,20));
JPanel nameTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
nameTextPanel.add(txtName);
textPanel.add(nameTextPanel);

txtAddress=new JTextField(20);
txtAddress.setFont(new Font("",1,20));
JPanel addressTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
addressTextPanel.add(txtAddress);
textPanel.add(addressTextPanel);

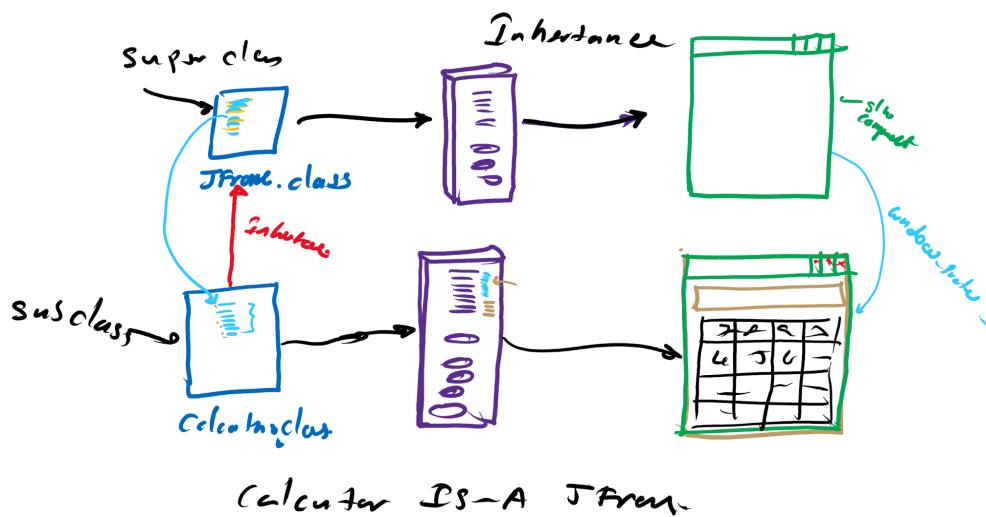
txtSalary=new JTextField(6);
txtSalary.setFont(new Font("",1,20));
JPanel salaryTextPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));
salaryTextPanel.add(txtSalary);
textPanel.add(salaryTextPanel);
add("Center",textPanel);

}
```

```
Q162
import java.util.ArrayList;
class CustomerController{
    public static boolean addCustomer(Customer customer){
        ArrayList <Customer>customerList=CustomerDBConnection.getInstance().getCustomerList();
        return customerList.add(customer);
    }
    public static boolean deleteCustomer(int index){
        CustomerDBConnection.getInstance().getCustomerList().remove(index);
        return true;
    }
    public static boolean deleteCustomer(Customer customer){
        return CustomerDBConnection.getInstance().getCustomerList().remove(customer);
    }
}
```

Inheritance

Q163 Case I



```
import javax.swing.*;
class Calculator extends JFrame{

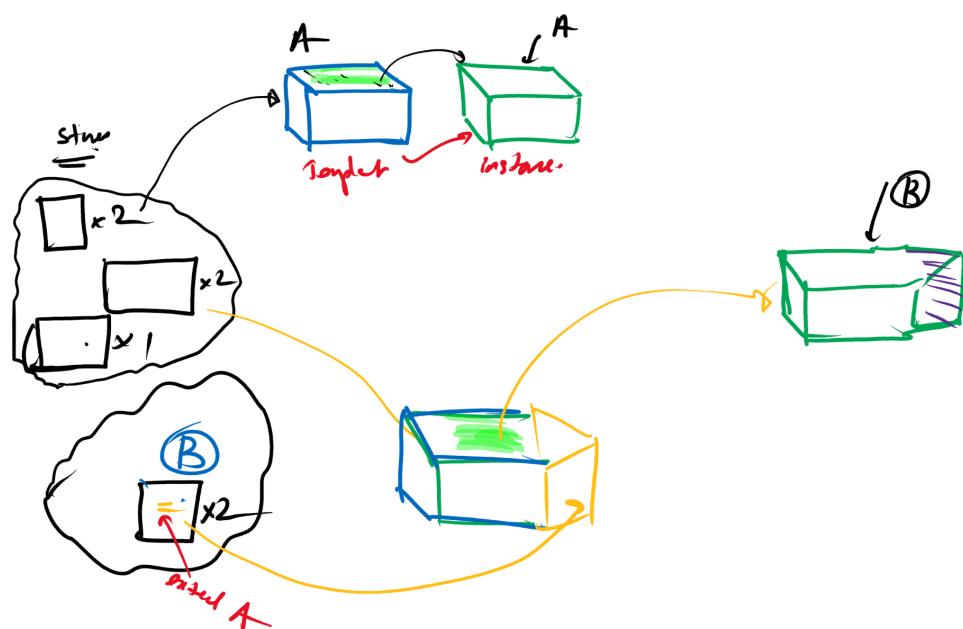
}

class Demo{
    public static void main(String args[]){
        Calculator c1=new Calculator();
        c1.setSize(300,300);
        c1.setTitle("Calculator");
        c1.setVisible(true);
    }
}

//Calculator IS-A JFrame
//Calculator->Subclass
//JFrame----->Superclass
```

```
///////////////////////////////
Q164 Case II
=====
import javax.swing.*;
class Calculator extends JFrame{
    Calculator(){
        setSize(300,300); //Legal ?
        setTitle("Calculator"); //Legal ?
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
class Demo{
    public static void main(String args[]){
        Calculator c1=new Calculator();
        c1.setVisible(true);
    }
}
```

```
///////////////////////////////
Q165 Case III
=====
```



```
class A{
    {
        System.out.println("Instance block A");
    }
}
class B extends A{
    {
        System.out.println("Instance block B");
    }
}
```

```
class C extends B{
    {
        System.out.println("Instance block C");
    }
}

class Demo{
    public static void main(String args[]){
        new C();
    }
}

///////////////////////////////
Q166
class A{
    {
        System.out.println("Instance block A");
    }
}

class B extends A{
    {
        System.out.println("Instance block B");
    }
}

class C extends B{
    {
        System.out.println("Instance block C");
    }
}

class D extends B{
    {
        System.out.println("Instance block D");
    }
}

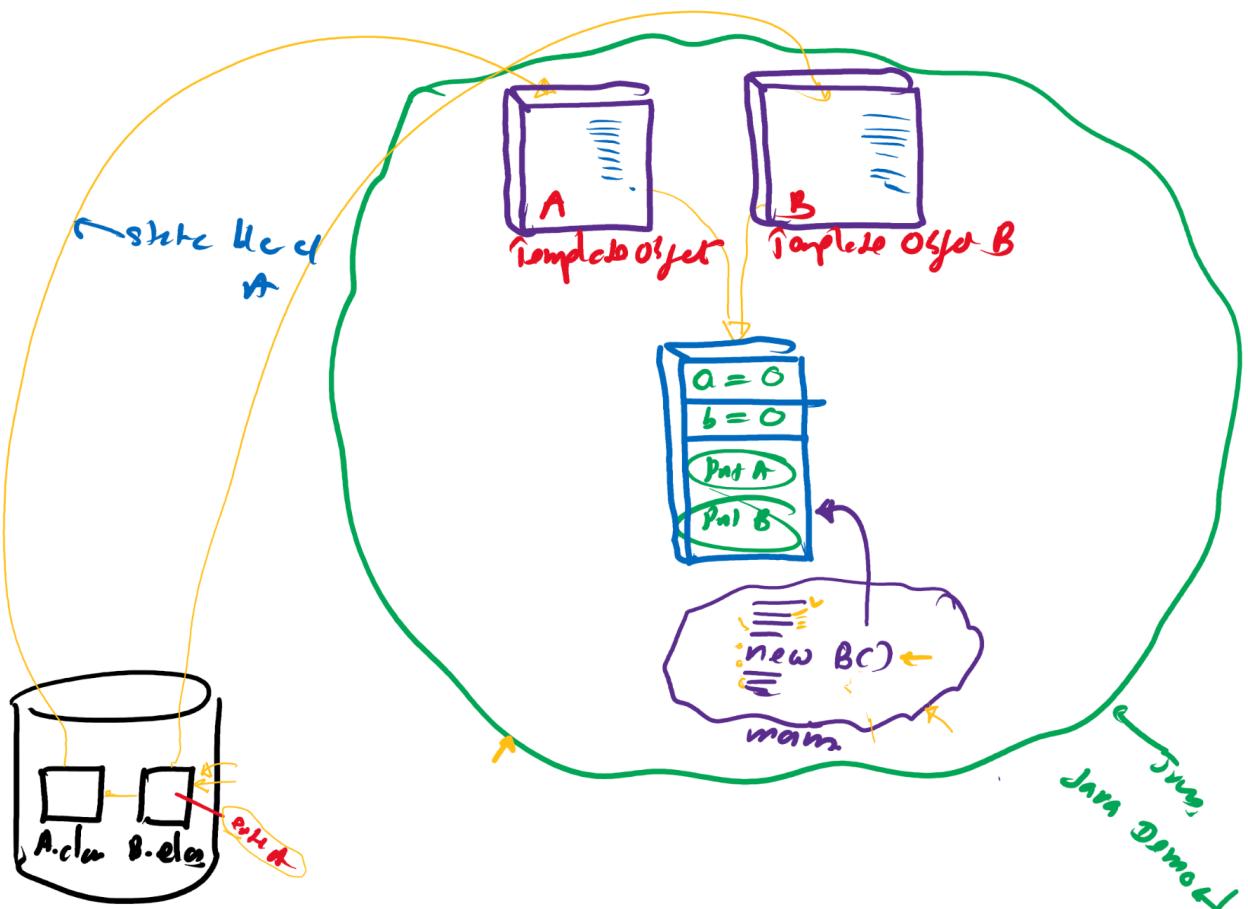
class Demo{
    public static void main(String args[]){
        new D();
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q167 Case IV

=====

```
class A{  
    int a;  
    public void printA(){  
        System.out.println("printA : "+a);  
    }  
}  
class B extends A{  
    int b;  
    public void printB(){  
        System.out.println("printB : "+b);  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        B b1=new B();  
    }  
}
```



||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q168 Case V

=====

```
class A{
    int a;
    public void printA(){
        System.out.println("printA : "+a);
    }
}
class B extends A{
    int b;
    public void printB(){
        System.out.println("printB : "+b);
    }
}
class Demo{
    public static void main(String args[]){
        B b1=null;
        b1.a=100;
        b1.b=200;
        b1.printA();
        b1.printB();
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q169 Case VI

=====

```
class A{
    int a;
    A(){
        System.out.println("A()");
    }
    A(int i){
        System.out.println("A(int)");
    }
    A(int i, int j){
        System.out.println("A(int,int)");
    }
}
class B extends A{
    int b;
    B(){
        System.out.println("B()");
    }
    B(int i){
        System.out.println("B(int)");
    }
    B(int i, int j){
        System.out.println("B(int,int)");
    }
}
```

```
class Demo{
    public static void main(String args[]){
        B b1=new B();
        System.out.println("-----");
        B b2=new B(100);
        System.out.println("-----");
        B b3=new B(100,200);
    }
}

///////////////////////////////
Q170
class A{
    int a;
    A(){
        System.out.println("A()");
    }
    A(int i){
        System.out.println("A(int)");
    }
    A(int i, int j){
        System.out.println("A(int,int)");
    }
}
class B extends A{
    int b;
    B(){
        super();
        System.out.println("B()");
    }
    B(int i){
        super();
        System.out.println("B(int)");
    }
    B(int i, int j){
        super();
        System.out.println("B(int,int)");
    }
}
class Demo{
    public static void main(String args[]){
        B b1=new B();
        System.out.println("-----");
        B b2=new B(100);
        System.out.println("-----");
        B b3=new B(100,200);
    }
}
```

```
////////////////////////////////////////////////////////////////////////
```

Q171

```
class A{
    int a;
    A(){
        System.out.println("A()");
    }
    A(int i){
        System.out.println("A(int)");
    }
    A(int i, int j){
        System.out.println("A(int,int)");
    }
}
class B extends A{
    int b;
    B(){
        super();
        System.out.println("B()");
    }
    B(int i){
        super(i);
        System.out.println("B(int)");
    }
    B(int i, int j){
        super(i,j);
        System.out.println("B(int,int)");
    }
}
```

}

```
class Demo{
    public static void main(String args[]){
        B b1=new B();
        System.out.println("-----");
        B b2=new B(100);
        System.out.println("-----");
        B b3=new B(100,200);
    }
}
```

```
////////////////////////////////////////////////////////////////////////
```

Q172 Exercise

=====

```
class A{
    int a;
    A(int i){
        System.out.println("A(int)");
    }
}
class B extends A{
    B(){
        super();
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q173 Case VII

=====

```
class A{  
}  
class B{  
  
}  
class C extends A,B{ //Illegal  
  
}
```

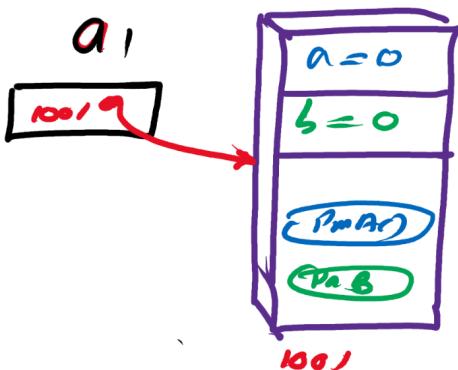
||||||||||||||||||||||||||||||||||||||||||||||||||||

Q174 Case VIII

=====

```
class A{  
    int a;  
    public void printA(){  
        System.out.println("printA : "+a);  
    }  
}  
class B extends A{  
    int b;  
    public void printB(){  
        System.out.println("printB : "+b);  
    }  
}  
class X{}  
class Demo{  
    public static void main(String args[]){  
        A a1=new B(); //Legal  
        a1.a=100; //Legal  
        a1.b=200; //Illegal  
        a1.printA(); //Legal  
        a1.printB(); //Illegal  
    }  
}
```

A a1 = new B()



a1.a=100; ✓
a1.printA(); ✓

a1.b=200; ✗
a1.printB(); ✗

//////////

Q175 Case IX (Method Overriding)

=====

```
class A{
    int a;
    public void printA(){
        System.out.println("printA : "+a);
    }
    public void print(){
        System.out.println("print() of A");
    }
}

class B extends A{
    int b;
    public void printB(){
        System.out.println("printB : "+b);
    }
    public void print(){ //Legal, Same name same signature->Method overriding
        System.out.println("print() of B");
    }
}

class Demo{
    public static void main(String args[]){
        B b1=new B();
        b1.print();
    }
}
```

//////////

Q176 Case X Method Overriding vs Overloading

=====

```
class A{
    int a;
    public void printA(){
        System.out.println("printA : "+a);
    }
    public void print(){
        System.out.println("print() of A");
    }
}

class B extends A{
    int b;
    public void printB(){
        System.out.println("printB : "+b);
    }
    public void print(){ //Method overriding
        System.out.println("print() of B");
    }
}
```

```
public void print(int length){ //Method Overloading
    System.out.println("print(int) of B");
}
}

class Demo{
    public static void main(String args[]){
        B b1=new B();
        b1.print();
        b1.print(100);
    }
}
```

//////////

Q177 Case XI

=====

```
class A{
    int a;
    public void print(){
        System.out.println("print() of A");
    }
}

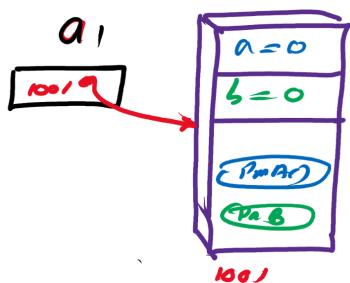
class B extends A{
    int b;
    public void print(){ //Method overriding
        System.out.println("print() of B");
    }
    public void callPrint(){
        print();
        super.print();
    }
}

class Demo{
    public static void main(String args[]){
        B b1=new B();
        b1.callPrint();
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q178 Case XII (Dynamic Method Dispatch)
=====
class A{
    int a;
    public void print(){
        //
    }
}
class B extends A{
    int b;
    public void print(){
        System.out.println("print() of B");
    }
}
class Demo{
    public static void main(String args[]){
        A a1=new B();
        a1.print();
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q179
class A{
    int a;
    public void print(){
        //
    }
}
class B extends A{
    int b;
    public void print(){
        System.out.println("print() of B");
    }
}
class C extends B{
    int c;
    public void print(){
        System.out.println("print() of C");
    }
}
class Demo{
    public static void main(String args[]){
        A a1=new C();
        a1.print();
    }
}
```

A a1 = new B();



a1.a=100; ✓
a2.print(); ✓
a1.b=200; ✗
a1.print(); ✗

//////////

Q180

```
class A{  
    int a;  
    public void print(){  
        System.out.println("print() of A");  
    }  
}  
class B extends A{  
    int b;  
    public void print(){  
        System.out.println("print() of B");  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        A a1=new A();  
        A a2=new B();  
        B b1=new B();  
  
        a1.print(); //?  
        a2.print(); //?  
        b1.print(); //?  
    }  
}
```

```
///////////////////////////////
Q181
class A{
    int a;
    public void print(){
        System.out.println("print() of A");
    }
    public void callPrint(){
        print();
    }
}
class B extends A{
    int b;
    public void print(){
        System.out.println("print() of B");
    }
}
class Demo{
    public static void main(String args[]){
        A a1=new A();
        A a2=new B();
        B b1=new B();

        a1.callPrint();
        a2.callPrint(); //ok
        b1.callPrint(); //ok
    }
}
```

```
///////////////////////////////
```

Polymorphism

Polymorphism (single interface many forms)

Compile time

- * Compile time
- * Compiler
- * Method Overloading
- * Static binding

Runtime Polymorphism

- * Runtime
- * JVM
- * Method Overriding
- * Dynamic binding

```
////////////////////////////////////////////////////////////////////////
Q182 Compile time polymorphism
=====
class Demo{
    public static void main(String args[]){
        int a=-122;
        long b=-2333;
        float c=-0.123f;
        double d=-12.12345;

        a=Math.abs(a);
        b=Math.abs(b);
        c=Math.abs(c);
        d=Math.abs(d);

        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        System.out.println(d);
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q183 Runtime Polymorphism
=====
class Car{
    public void park(){
        System.out.println("Car Parking...");
    }
}
class Bus{
    public void park(){
        System.out.println("Bus Parking...");
    }
}
class Van{
    public void park(){
        System.out.println("Van Parking...");
    }
}
class Demo{
    public static void main(String args[]){
        ob[]={new Car(),new Bus(),new Van()};
        //call "park()"
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q184 From Q183
=====
class Vehicle{

}

class Car extends Vehicle{
    public void park(){
        System.out.println("Car Parking...");
    }
}

class Bus extends Vehicle{
    public void park(){
        System.out.println("Bus Parking...");
    }
}

class Van extends Vehicle{
    public void park(){
        System.out.println("Van Parking...");
    }
}

class Demo{
    public static void main(String args[]){
        Vehicle ob[]={new Car(),new Bus(),new Van()};
        //call "park()"
    }
}
```

```
////////////////////////////////////////////////////////////////////////
Q185 From Q184
=====
class Vehicle{

}

class Car extends Vehicle{
    public void park(){
        System.out.println("Car Parking...");
    }
}

class Bus extends Vehicle{
    public void park(){
        System.out.println("Bus Parking...");
    }
}

class Van extends Vehicle{
    public void park(){
        System.out.println("Van Parking...");
    }
}
```

```
class Demo{
    public static void main(String args[]){
        Vehicle vehicle[]={new Car(),new Bus(),new Van()};
        //call "park()"
        for(Vehicle v1 : vehicle){
            v1.park(); //Illegal
        }
    }
}
```

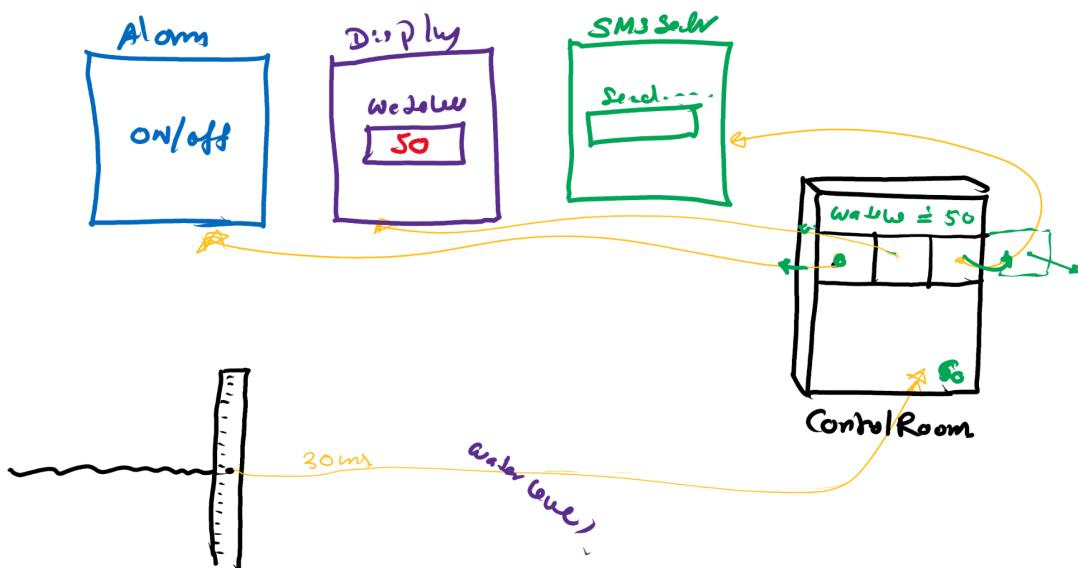
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q186 From Q185

```
=====
class Vehicle{
    public void park(){
        System.out.println("Vehicle Parking...");
    }
}
class Car extends Vehicle{
    public void park(){
        System.out.println("Car Parking...");
    }
}
class Bus extends Vehicle{
    public void park(){
        System.out.println("Bus Parking...");
    }
}
class Van extends Vehicle{
    public void park(){
        System.out.println("Van Parking...");
    }
}
class Demo{
    public static void main(String args[]){
        Vehicle vehicle[]={new Car(),new Bus(),new Van()};
        //call "park()"
        for(Vehicle v1 : vehicle){
            v1.park(); //Legal
        }
    }
}
```

||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q187



||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||

Q188

```
import java.util.*;
class Alarm{
    public void operateAlarm(int waterLevel){
        System.out.println(waterLevel>=50 ? "Alarm ON":"Alarm OFF");
    }
}
class Display{
    public void display(int waterLevel){
        System.out.println("Water Level : "+waterLevel);
    }
}
class SMSSender{
    public void send(int waterLevel){
        System.out.println("sending : "+waterLevel);
    }
}
class ControlRoom{
    private int waterLevel;
    private Alarm alarm;
    private Display display;
    private SMSSender sMSSender;

    public void addAlarm(Alarm alarm){
        this.alarm=alarm;
    }
    public void addDisplay(Display display){
        this.display=display;
    }
    public void addSMSSender(SMSSender sMSSender){
        this.sMSSender=sMSSender;
    }
    public void setWaterLevel(int waterLevel){
```

```

        if(this.waterLevel!=waterLevel){
            this.waterLevel=waterLevel;
            notifyObject();
        }
    }
    public void notifyObject(){
        alarm.operateAlarm(waterLevel);
        display.display(waterLevel);
        sMSSender.send(waterLevel);
    }
}

class Demo{
    public static void main(String args[]){
        ControlRoom conRoom=new ControlRoom();
        conRoom.addAlarm(new Alarm());
        conRoom.addDisplay(new Display());
        conRoom.addSMSender(new SMSender());

        Random r=new Random();
        while(true){
            int waterLevel=r.nextInt(101);
            conRoom.setWaterLevel(waterLevel);
            try{Thread.sleep(1000);}catch(InterruptedException ex){}
        }
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

Q189 Adding a Splitter

```

import java.util.*;
class Alarm{
    public void operateAlarm(int waterLevel){
        System.out.println(waterLevel>=50 ? "Alarm ON":"Alarm OFF");
    }
}
class Splitter{
    public void split(int waterLevel){
        System.out.println(waterLevel>=75 ? "Splitter ON":"Splitter OFF");
    }
}
class Display{
    public void display(int waterLevel){
        System.out.println("Water Level : "+waterLevel);
    }
}
class SMSender{
    public void send(int waterLevel){
        System.out.println("sending : "+waterLevel);
    }
}

```

```
class ControlRoom{
    private int waterLevel;
    private Alarm alarm;
    private Display display;
    private SMSSender sMSSender;

    private Splitter splitter;

    public void addSplitter(Splitter splitter){
        this.splitter=splitter;
    }
    public void addAlarm(Alarm alarm){
        this.alarm=alarm;
    }
    public void addDisplay(Display display){
        this.display=display;
    }
    public void addSMSSender(SMSSender sMSSender){
        this.sMSSender=sMSSender;
    }
    public void setWaterLevel(int waterLevel){
        if(this.waterLevel!=waterLevel){
            this.waterLevel=waterLevel;
            notifyObject();
        }
    }
    public void notifyObject(){
        alarm.operateAlarm(waterLevel);
        display.display(waterLevel);
        sMSSender.send(waterLevel);
        splitter.split(waterLevel);
    }
}

class Demo{
    public static void main(String args[]){
        ControlRoom conRoom=new ControlRoom();
        conRoom.addAlarm(new Alarm());
        conRoom.addDisplay(new Display());
        conRoom.addSMSSender(new SMSSender());
        conRoom.addSplitter(new Splitter());

        Random r=new Random();
        while(true){
            int waterLevel=r.nextInt(101);
            conRoom.setWaterLevel(waterLevel);
            try{Thread.sleep(1000);}catch(InterruptedException ex){}
        }
    }
}
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Q190 From Q188 Using “Polymorphism”

```
=====
```

```
import java.util.*;  
class WaterLevelObserver{  
    public void update(int waterLevel){  
  
    }  
}  
class Alarm extends WaterLevelObserver{  
    public void update(int waterLevel){  
        System.out.println(waterLevel>=50 ? "Alarm ON":"Alarm OFF");  
    }  
}  
class Display extends WaterLevelObserver{  
    public void update(int waterLevel){  
        System.out.println("Water Level : "+waterLevel);  
    }  
}  
class SMSSender extends WaterLevelObserver{  
    public void update(int waterLevel){  
        System.out.println("sending : "+waterLevel);  
    }  
}  
class ControlRoom{  
    private int waterLevel;  
  
    private WaterLevelObserver[] observerArray=new WaterLevelObserver[100];  
    private int nextIndex;  
  
    public void addWaterLevelObserver(WaterLevelObserver ob){  
        observerArray[nextIndex++]=ob;  
    }  
  
    public void setWaterLevel(int waterLevel){  
        if(this.waterLevel!=waterLevel){  
            this.waterLevel=waterLevel;  
            notifyObject();  
        }  
    }  
    public void notifyObject(){  
        for(int i=0; i<nextIndex; i++){  
            observerArray[i].update(waterLevel);  
        }  
    }  
}  
class Demo{  
    public static void main(String args[]){  
        ControlRoom conRoom=new ControlRoom();  
        conRoom.addWaterLevelObserver(new Alarm());
```

```

        conRoom.addWaterLevelObserver(new Display());
        conRoom.addWaterLevelObserver(new SMSSender());

        Random r=new Random();
        while(true){
            int waterLevel=r.nextInt(101);
            conRoom.setWaterLevel(waterLevel);
            try{Thread.sleep(1000);}catch(InterruptedException ex){}
        }
    }

=====

```

Q191 Adding a Splitter

```

=====
import java.util.*;
class WaterLevelObserver{
    public void update(int waterLevel){

    }
}

class Alarm extends WaterLevelObserver{
    public void update(int waterLevel){
        System.out.println(waterLevel>=50 ? "Alarm ON":"Alarm OFF");
    }
}

class Splitter extends WaterLevelObserver{
    public void update(int waterLevel){
        System.out.println(waterLevel>=75 ? "Splitter ON":"Splitter OFF");
    }
}

class Display extends WaterLevelObserver{
    public void update(int waterLevel){
        System.out.println("Water Level : "+waterLevel);
    }
}

class SMSSender extends WaterLevelObserver{
    public void update(int waterLevel){
        System.out.println("sending : "+waterLevel);
    }
}

class ControlRoom{
    private int waterLevel;

    private WaterLevelObserver[] observerArray=new WaterLevelObserver[100];
    private int nextIndex;

    public void addWaterLevelObserver(WaterLevelObserver ob){
        observerArray[nextIndex++]=ob;
    }
}
```

```

public void setWaterLevel(int waterLevel){
    if(this.waterLevel!=waterLevel){
        this.waterLevel=waterLevel;
        notifyObject();
    }
}

public void notifyObject(){
    for(int i=0; i<nextIndex; i++){
        observerArray[i].update(waterLevel);
    }
}

}

class Demo{
    public static void main(String args[]){
        ControlRoom conRoom=new ControlRoom();
        conRoom.addWaterLevelObserver(new Alarm());
        conRoom.addWaterLevelObserver(new Display());
        conRoom.addWaterLevelObserver(new SMSSender());
        conRoom.addWaterLevelObserver(new Splitter());

        Random r=new Random();
        while(true){
            int waterLevel=r.nextInt(101);
            conRoom.setWaterLevel(waterLevel);
            try{Thread.sleep(1000);}catch(InterruptedException ex){}
        }
    }
}

```

Q192 From Q191

```

import java.util.*;
class WaterLevelObserver{
    public void update(int waterLevel){

    }
}

class Alarm extends WaterLevelObserver{
    public void update(int waterLevel){
        System.out.println(waterLevel>=50 ? "Alarm ON":"Alarm OFF");
    }
}

class Splitter extends WaterLevelObserver{
    public void update(int waterLevel){
        System.out.println(waterLevel>=75 ? "Splitter ON":"Splitter OFF");
    }
}

class Display extends WaterLevelObserver{

```

```

public void update(int waterLevel){
    System.out.println("Water Level : "+waterLevel);
}

}

class SMSSender extends WaterLevelObserver{
    public void update(int waterLevel){
        System.out.println("sending : "+waterLevel);
    }
}

class ControlRoom{
    private int waterLevel;

    private ArrayList<WaterLevelObserver>observerList=new ArrayList<>();

    public void addWaterLevelObserver(WaterLevelObserver ob){
        observerList.add(ob);
    }

    public void setWaterLevel(int waterLevel){
        if(this.waterLevel!=waterLevel){
            this.waterLevel=waterLevel;
            notifyObject();
        }
    }

    public void notifyObject(){
        for(WaterLevelObserver ob : observerList){
            ob.update(waterLevel);
        }
    }
}

}

class Demo{
    public static void main(String args[]){
        ControlRoom conRoom=new ControlRoom();
        conRoom.addWaterLevelObserver(new Alarm());
        conRoom.addWaterLevelObserver(new Display());
        conRoom.addWaterLevelObserver(new SMSSender());
        conRoom.addWaterLevelObserver(new Splitter());

        Random r=new Random();
        while(true){
            int waterLevel=r.nextInt(101);
            conRoom.setWaterLevel(waterLevel);
            try{Thread.sleep(1000);}catch(InterruptedException ex){}
        }
    }
}

s
///////////
Q193

```

Q195

Q196

Q197

||||| Q198

||||| Q200