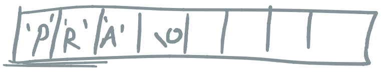
String and Vector

13 October 2022 07:01 PM



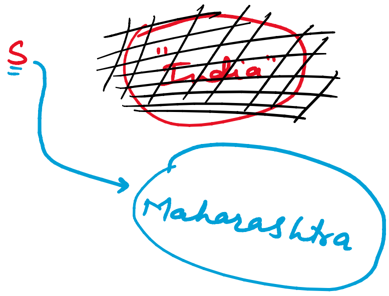
class → Object → methods .

String s =new String("India");

→ String s = "India";

S = "Maharashtra";

→ Immutable .



String s1 = S.toLowerCase();

maharastra

S = S.toLowerCase();

1) length();
2) toLowerCase()
3) toUpperCase()
4) trim()
5) replace(char,char);
6) equals( ) India ≠ india
7) equalsIgnoreCase() India = india
8) substring(a) → 2, dia
9) substring(n1,n2) ;(1,3) "nd"

10) concat()
11) charAt() → 3 → 'i'
12) charAt(char,int);
13) toCharArray();

```
import java.util.*;
public class Main
{
        public static void main(String[] args) {
           Scanner s=new Scanner(System.in);
                System.out.println("Enter a string");
                String s1=s.nextLine();
                System.out.println("Original String is = "+s1);
                StringBuffer sb=new StringBuffer();
                sb.append(s1);
                sb.reverse();
                System.out.println("StringBuffer is = "+sb);
                String rev=sb.toString();
                System.out.println("String Reverse is = "+rev);
                if(s1.equalsIgnoreCase(rev))
                System.out.println("String is Palindrome ");
                else
                System.out.println("String is not a Palindrome ");
        }
}
```

StringBuffer . → Mutable

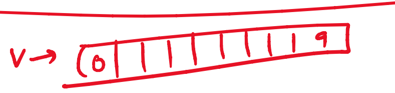
1) length()
2) setCharAt(char,int)
3) charAt()
4) insert(int,char)
5) delete(int,int)
6) append()
7) reverse()
8) toString()

Vector .

Vector v=new Vector();

Vector v=new Vector(5,3);

Vector v=new Vector(6);



1) addElement()
2) firstElement()
3) lastElement()
4) removeElement()

```
import java.util.*;
public class Main
{
           public static void main(String[] args) {
              Vector v=new Vector(5,2);
              v.addElement(new Integer(1));
              v.addElement(new Double(7.8));
              v.addElement(new Float(3.2));
              v.addElement(new Long(1000045));
              v.addElement(new String("DJSCE"));
                      System.out.println("Vector is "+v);
```

3) lastElement()
4) removeElement()
5) indexOf()
6) lastIndexOf()
7) elementAt()
8) removeElementAt()

```
v.addElement(new Float(3.2));
v.addElement(new Long(1000045));
v.addElement(new String("DJSCE"));
        System.out.println("Vector is "+v);
        System.out.println("Capacity is = "+v.capacity());
        System.out.println("Size is = "+v.size());
        v.addElement(new String("Sixth Element"));
        System.out.println("Vector is "+v);
        System.out.println("Capacity is = "+v.capacity());
        System.out.println("Size is = "+v.size());
    }
}
```

Constructors-

```
class Circle
{ float a,r;  Circle(int x)
              { r=x;
              }
   void process()
   { a=3.14f*r*r;
   }
   void output()
   {
     Sop("Area is"+a);
   }
}
```

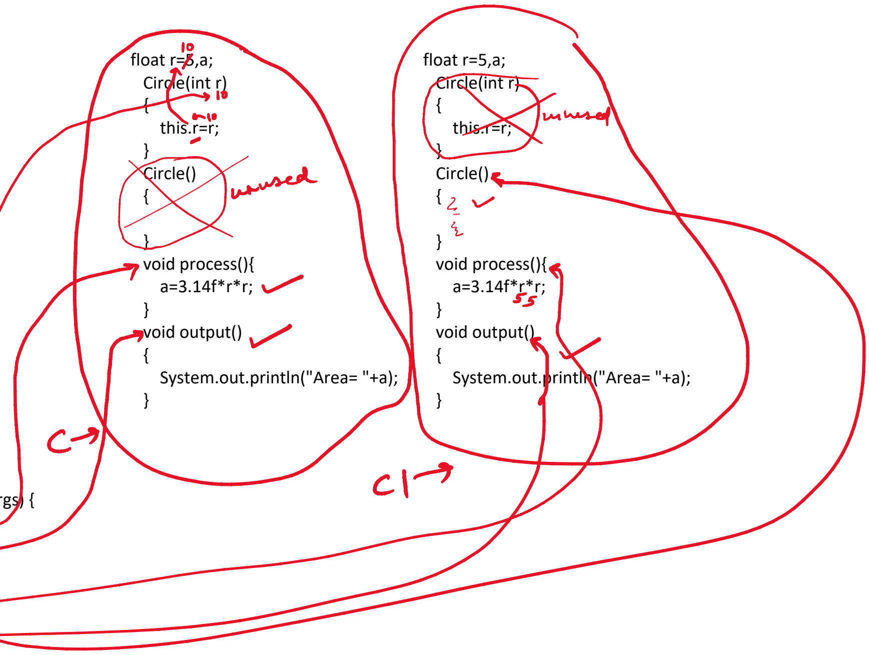
Circle c=new Circle(10);

this keyword

Constructor Overloading.

```
//import java.util.*;
class Circle
{
   float r=5,a;
   Circle(int r)
   {
      this.r=r;
   }
   Circle()
   {

   }
   void process(){
      a=3.14f*r*r;
   }
   void output()
   {
      System.out.println("Area= "+a);
   }
}
public class Main
{
        public static void main(String[] args) {
           Circle c=new Circle(10);
           c.process();
           c.output();
            Circle c1=new Circle();
           c1.process();
           c1.output();
        }
}
```



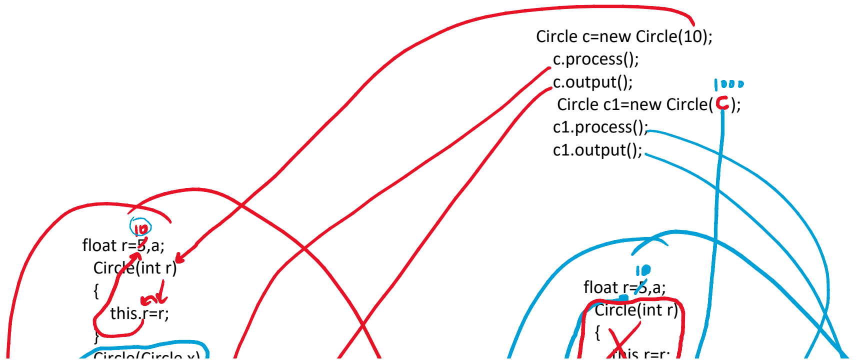
```
float r=5,a;
  Circle(int r)
  {
     this.r=r;
  }
  Circle()
  {

  }
  void process(){
     a=3.14f*r*r;
  }
  void output()
  {
     System.out.println("Area= "+a);
  }
```

```
float r=5,a;
  Circle(int r)
  {
     this.r=r;
  }
  Circle()
  {

  }
  void process(){
     a=3.14f*r*r;
  }
  void output()
  {
     System.out.println("Area= "+a);
  }
```

Copy Constructor.

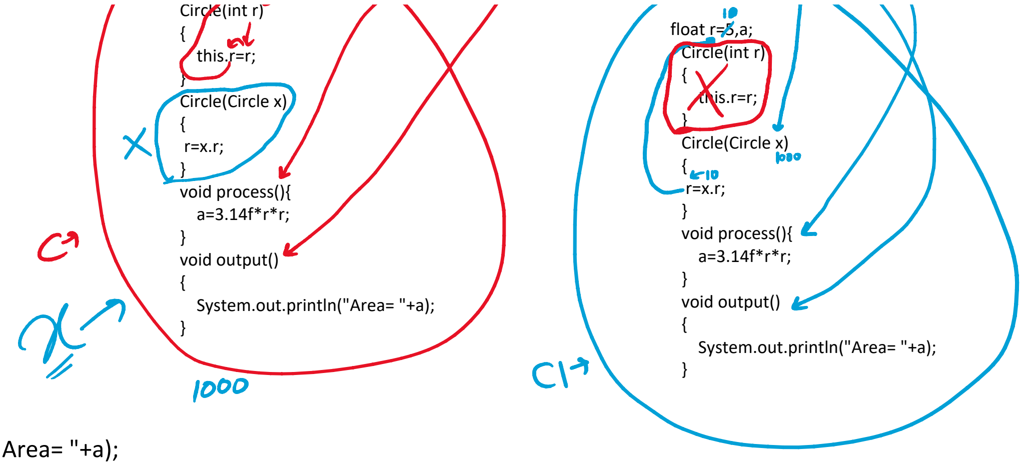


```
Circle c=new Circle(10);
  c.process();
  c.output();
   Circle c1=new Circle(c);
  c1.process();
  c1.output();
```

```
float r=5,a;
  Circle(int r)
  {
     this.r=r;
  }
  Circle(Circle x)
```

```
float r=5,a;
  Circle(int r)
  {
     this.r=r;
```

```
class Circle
{
   float r=5,a;
   Circle(int r)
   {
    float r=5,a;
    Circle(int r)
    {
        this.r=r;
    }
    Circle(Circle x)
    {
     r=x.r;
    }
    void process(){
        a=3.14f*r*r;
    }
    void output()
    {
        System.out.println("Area= "+a);
    }
}
public class Main
{
        public static void main(String[] args) {
            Circle c=new Circle(10);
            c.process();
            c.output();
             Circle c1=new Circle(c);
            c1.process();
            c1.output();
        }
}
```



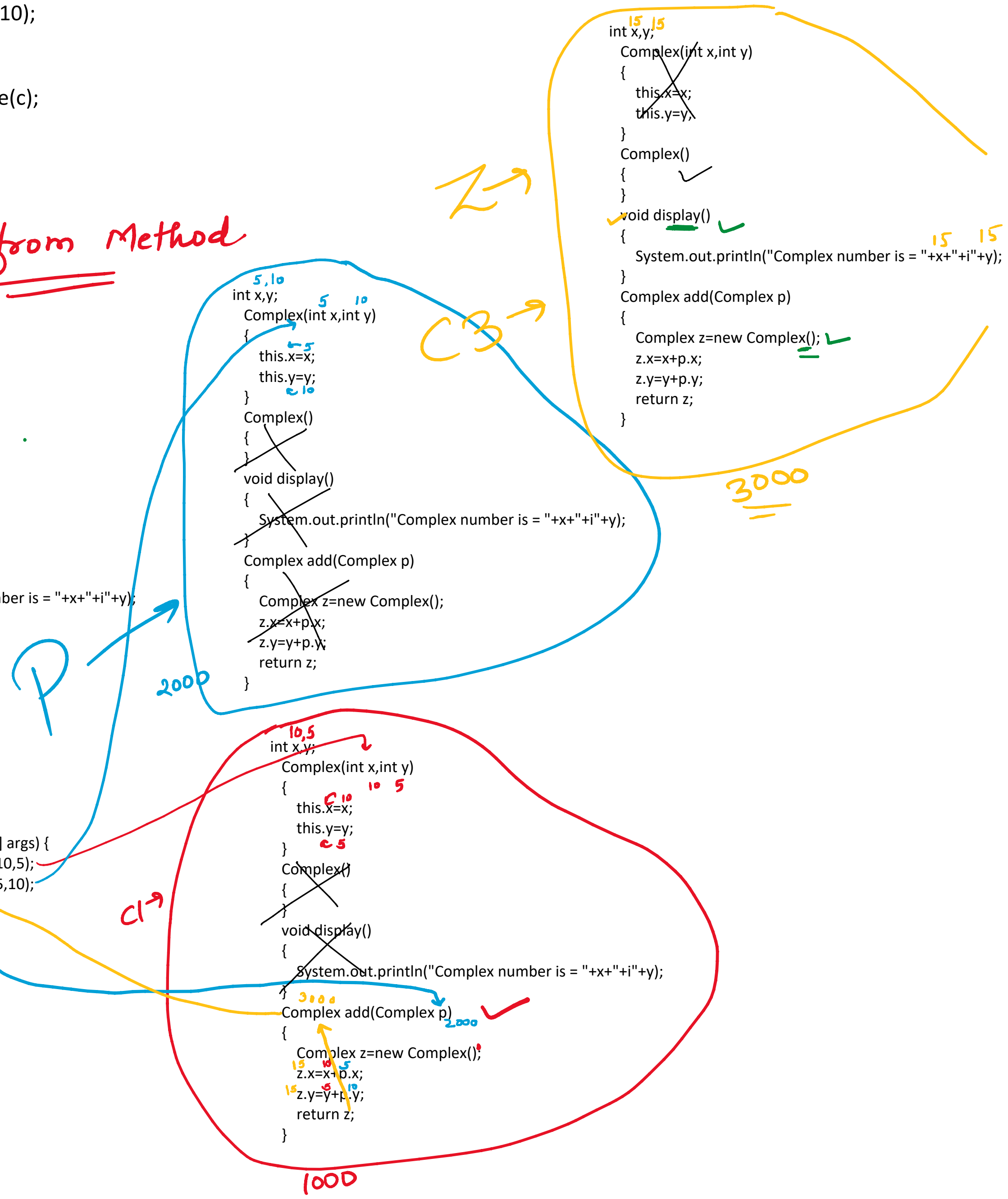
```
Circle(int r)
{
 this.r=r;
}
Circle(Circle x)
{
 r=x.r;
}
void process(){
   a=3.14f*r*r;
}
void output()
{
   System.out.println("Area= "+a);
}
```

```
float r=5,a;
Circle(int r)
{
  this.r=r;
}
Circle(Circle x)
{
 r=x.r;
}
void process(){
   a=3.14f*r*r;
}
void output()
{
   System.out.println("Area= "+a);
}
```

Returning Object from Method

```
class Complex
{
   int x,y;
   Complex(int x,int y)
   {
      this.x=x;
      this.y=y;
   }
   Complex()
   {
   }
   void display()
   {
      System.out.println("Complex number is = "+x+"i"+y);
   }
   Complex add(Complex p)
   {
      Complex z=new Complex();
      z.x=x+p.x;
      z.y=y+p.y;
      return z;
   }
}
public class Main
{
        public static void main(String[] args) {
           Complex c1=new Complex(10,5);
           Complex c2=new Complex(5,10);
           Complex c3=c1.add(c2);
           c3.display();
        }
}
```

15+i15



```
int x,y;
 Complex(int x,int y)
 {
    this.x=x;
    this.y=y;
 }
 Complex()
 {
 }
 void display()
 {
    System.out.println("Complex number is = "+x+"i"+y);
 }
 Complex add(Complex p)
 {
    Complex z=new Complex();
    z.x=x+p.x;
    z.y=y+p.y;
    return z;
 }
```

```
int x,y;
 Complex(int x,int y)
 {
    this.x=x;
    this.y=y;
 }
 Complex()
 {
 }
 void display()
 {
    System.out.println("Complex number is = "+x+"i"+y);
 }
 Complex add(Complex p)
 {
    Complex z=new Complex();
    z.x=x+p.x;
    z.y=y+p.y;
    return z;
 }
```

```
int x,y;
 Complex(int x,int y)
 {
    this.x=x;
    this.y=y;
 }
 Complex()
 {
 }
 void display()
 {
    System.out.println("Complex number is = "+x+"i"+y);
 }
 Complex add(Complex p)
 {
    Complex z=new Complex();
    z.x=x+p.x;
    z.y=y+p.y;
    return z;
 }
```