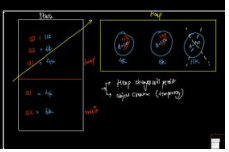
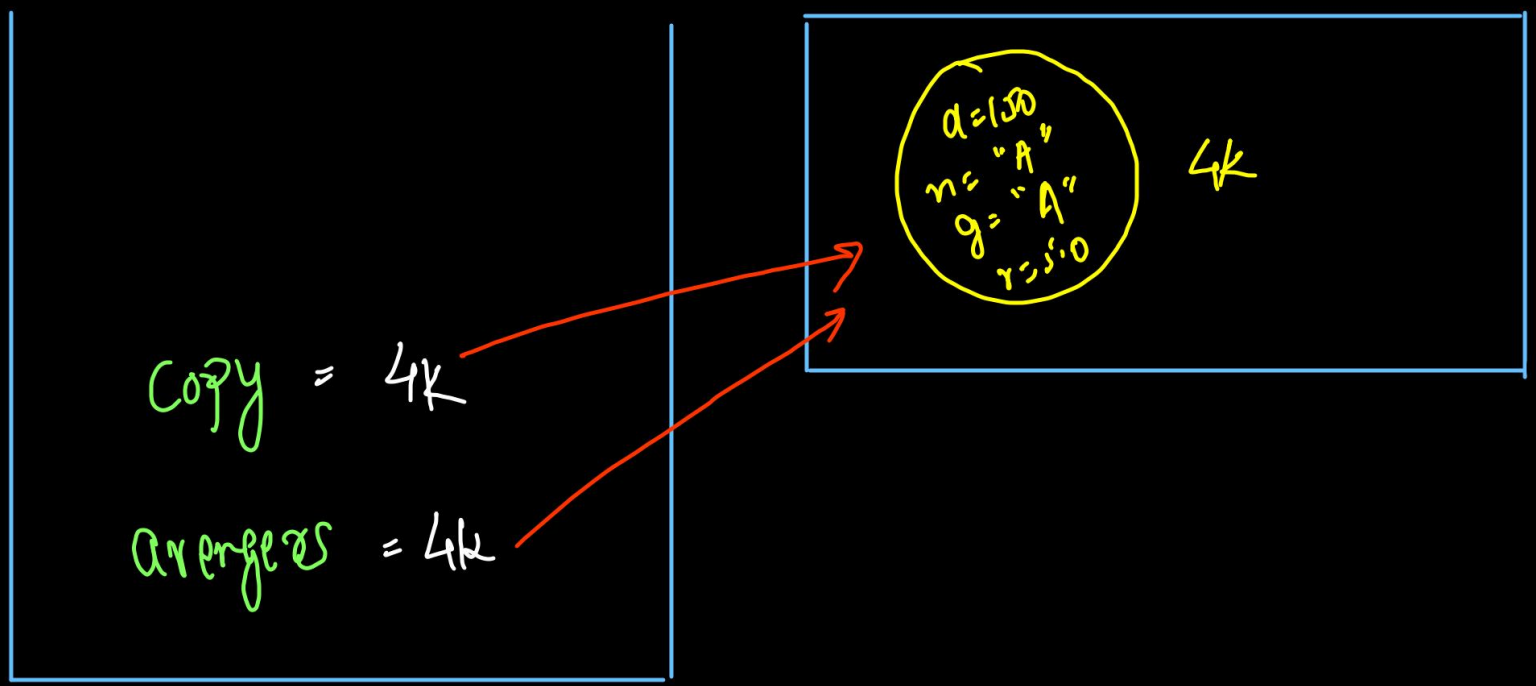
Q) What is the difference between shallow copy and deep copy?

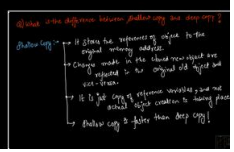
Shallow Copy :-

- It stores the references of object to the original memory address.
- Changes made in the cloned new object are reflected in the original old object and vice-versa.
- It is just copy of reference variables, and not actual object creation is taking place
- Shallow copy is faster than deep copy!

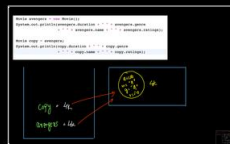


```java
Movie avengers = new Movie();
System.out.println(avengers.duration + " " + avengers.genre
                    + " " + avengers.name + " " + avengers.ratings);

Movie copy = avengers;
System.out.println(copy.duration + " " + copy.genre
                    + " " + copy.name + " " + copy.ratings);
```





```java
class Movie{
    int duration = 150;
    String genre = "Action", name = "Avengers";
    double ratings = 5.0;
}

public class Main {
    public static void main(String[] args) {
        Movie avengers = new Movie();
        System.out.println(avengers.duration + " " + avengers.genre
                           + " " + avengers.name + " " + avengers.ratings);

        Movie copy = avengers;
        System.out.println(copy.duration + " " + copy.genre
                           + " " + copy.name + " " + copy.ratings);

        copy.duration = 180;
        System.out.println(avengers.duration + " " + avengers.genre
                           + " " + avengers.name + " " + avengers.ratings);
        System.out.println(copy.duration + " " + copy.genre
                           + " " + copy.name + " " + copy.ratings);
    }
```



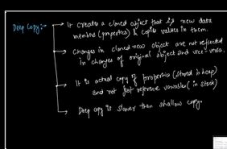
Deep Copy :→

- It creates a cloned object that is new data members (properties) & copies values in them.
- Changes in cloned new object are not reflected in changes of original object and vice-versa.
- It is actual copy of properties (stored in heap) and not just reference variables (in stack)
- Deep copy is slower than shallow copy.

```
    Movie(Movie other){
        duration = other.duration;
        genre = other.genre;
        name = other.name;
        ratings = other.ratings;
    }
}

public class Main {
    public static void main(String[] args) {
        Movie avengers = new Movie();
        System.out.println(avengers.duration + " " + avengers.genre
                           + " " + avengers.name + " " + avengers.ratings);

        Movie deepCopy = new Movie(avengers);
        System.out.println(deepCopy.duration + " " + deepCopy.genre
                           + " " + deepCopy.name + " " + deepCopy.ratings);

        avengers.duration = 180;
        System.out.println(avengers.duration + " " + avengers.genre
                           + " " + avengers.name + " " + avengers.ratings);
        System.out.println(deepCopy.duration + " " + deepCopy.genre
                           + " " + deepCopy.name + " " + deepCopy.ratings);
```

deepCopy = 6k

avengers = 4k



```java
class Movie {
    int duration;
    String name, genre;
    double rating;
    ArrayList<String> languages;

    public Movie(String name, int duration, double rating, String genre) {
        this.name = name;
        this.duration = duration;
        this.genre = genre;
        this.rating = rating;
        this.languages = new ArrayList<>();
    }

    // PARTIAL DEEP COPY CONSTRUCTOR
    public Movie(Movie other) {
        this.name = other.name;
        this.duration = other.duration;
        this.genre = other.genre;
        this.rating = other.rating;
        this.languages = other.languages;
    }
}
```

```
[English, Hindi]
architaggarwal@Archits-MacBook-Air Java OOPS % javac OOPS_Codes/6.1.ShallowCopy.java
architaggarwal@Archits-MacBook-Air Java OOPS % java OOPS_Codes.Driver
Avengers Endgame 180 4.5 SuperHero
[English]
[English, Tamil, Malayanam]
[English, Tamil, Malayanam]
Avengers Endgame 180 4.5 SuperHero
[English, Tamil, Malayanam]
[English, Tamil, Malayanam, Hindi, Telugu]
[English, Tamil, Malayanam, Hindi, Telugu]
architaggarwal@Archits-MacBook-Air Java OOPS %
```

```java
class Driver {
    Run | Debug
    public static void main(String[] args) {

        Movie avengers = new Movie(name: "Avengers Endgame", duration: 180,
                rating: 4.5, genre: "SuperHero");
        avengers.languages.add(e: "English");

        System.out.println(avengers.name + " " + avengers.duration
                + " " + avengers.rating + " " + avengers.genre);
        System.out.println(avengers.languages);

        // SHALLOW COPY

        Movie shallowCopy = avengers;

        avengers.languages.add(e: "Tamil");
        shallowCopy.languages.add(e: "Malayanam");

        System.out.println(shallowCopy.languages);
        System.out.println(avengers.languages);

        // PARTIAL DEEP COPY

        Movie partialDeep = new Movie(avengers);
        System.out.println(partialDeep.name + " " + partialDeep.duration
                + " " + partialDeep.rating + " " + partialDeep.genre);
        System.out.println(partialDeep.languages);

        avengers.languages.add(e: "Hindi");
        partialDeep.languages.add(e: "Telugu");

        System.out.println(partialDeep.languages);
        System.out.println(avengers.languages);
    }
}
```

```java
class Movie {
    int duration;
    String name, genre;
    double rating;
    ArrayList<String> languages;

    public Movie(String name, int duration, double rating, String genre) {
        this.name = name;
        this.duration = duration;
        this.genre = genre;
        this.rating = rating;
        this.languages = new ArrayList<>();
    }

    // COMPLETE DEEP COPY
    public Movie(Movie other) {
        this.name = other.name;
        this.duration = other.duration;
        this.genre = other.genre;
        this.rating = other.rating;
        this.languages = new ArrayList<>();
        for (String language : other.languages) {
            this.languages.add(language);
        }
    }
}
```

```java
class Driver {
    Run | Debug
    public static void main(String[] args) {

        Movie avengers = new Movie(name: "Avengers Endgame", duration: 180, rating: 4.5,
        genre: "SuperHero");
        avengers.languages.add(e: "English");

        System.out.println(avengers.name + " " + avengers.duration
                + " " + avengers.rating + " " + avengers.genre);
        System.out.println(avengers.languages);

        // PROPER DEEP COPY

        Movie deepCopy = new Movie(avengers);
        System.out.println(deepCopy.name + " " + deepCopy.duration
                + " " + deepCopy.rating + " " + deepCopy.genre);
        System.out.println(deepCopy.languages);

        avengers.languages.add(e: "Hindi");
        deepCopy.languages.add(e: "Telugu");

        System.out.println(deepCopy.languages);
        System.out.println(avengers.languages);
    }
}
```

```
architaggarwal@Archits-MacBook-Air Java OOPS % javac OOPS_Codes/6.2.DeepCopy.java
architaggarwal@Archits-MacBook-Air Java OOPS % java OOPS_Codes.Driver
Avengers Endgame 180 4.5 SuperHero
[English]
Avengers Endgame 180 4.5 SuperHero
[English]
[English, Telugu]
[English, Hindi]
```

Q) What are differences between methods & functions in Java ?

A method is a function or procedure in object oriented programming.

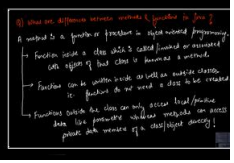
- Function inside a class which is called / invoked or associated with objects of that class is known as a method.
- Functions can be written inside as well as outside classes. ie. functions do not need a class to be created.
- Functions outside the class can only access local / primitive data like parameters whereas methods can access private data members of a class / object directly !

a) What is this keyword? What are it's applications?

THIS Keyword ⇝ Self referential pointer (reference variable pointing to current object)

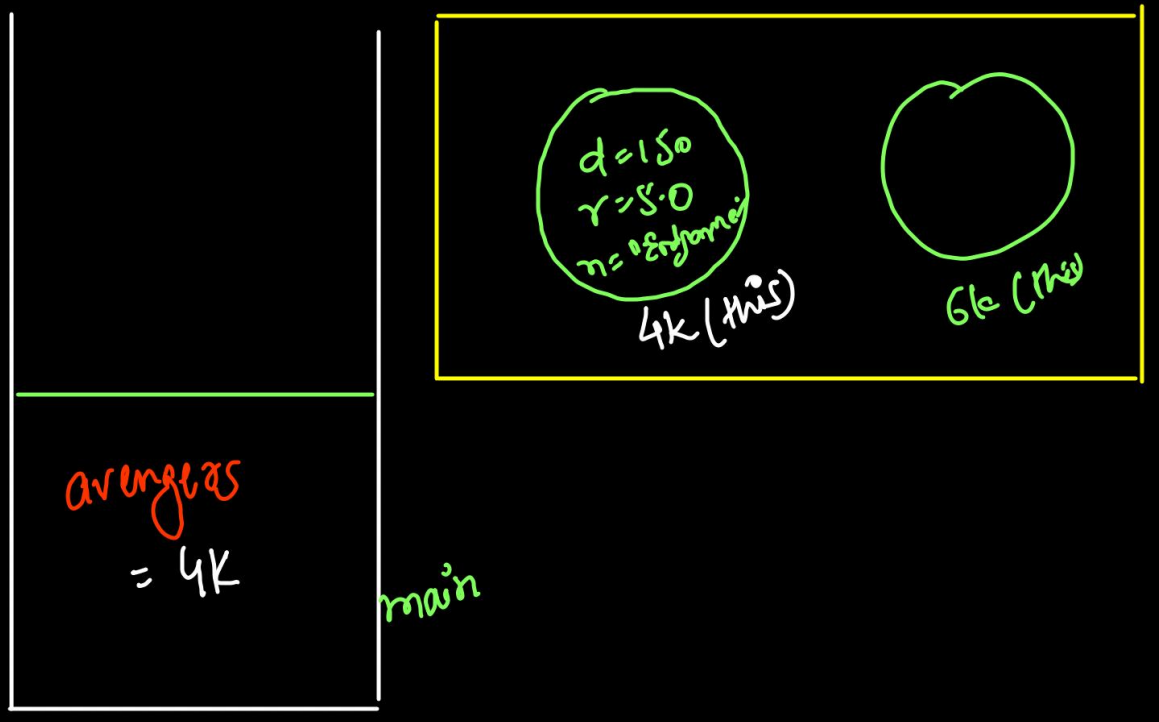
APPLICATIONS ⇒

1) → Refer data members from within the class → Resolve name conflict b/w parameters & data members
2) → Refer member functions from within the class
3) → Used in constructor chaining (invoke own constructor) → chaining should be the first & single call only!
4) → Pass the current object as a parameter to a method/constructor
5) → Return the current object from a method/constructor → Return type of a constructor

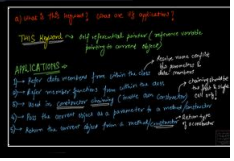


```java
class Movie {
    int duration;
    double ratings;
    String name;
    String genre;

    public Movie() {
    }

    public Movie(int newDuration, double newRatings,
                 String newName, String newGenre) {
        duration = newDuration;
        ratings = newRatings;
        name = newName;
        genre = newGenre;
    }
}
```

IMPLICIT RETURN TYPE: THIS (MOVIE REFERENCE TYPE)




```java
public static void main(String[] args) {
    Movie avengers = new Movie(newDuration: 150, newRatings: 5.0, newName: "Avengers Endgame",
    newGenre: "Thriller");
    System.out.println(avengers.duration + " " + avengers.name
            + " " + avengers.ratings + " " + avengers.genre);
}
```



```java
class Movie {
    private int duration;

    // Application 1: Constructor Return type
    // Application 2: Invoking Member Function
    public Movie(int duration) {
        this.setDuration(duration);
    }

    public int getDuration() {
        return duration;
    }

    // Application 3: Access Data Member Properties
    public void setDuration(int duration) {
        this.duration = duration;
    }

    // Application 4: Pass Current Object as Parameter
    public void display() {
        Driver.displayDurationOutside(this);
    }

    // Application 5: Return Current Object
    public Movie join(Movie other) {
        this.duration += other.duration;
        return this;
    }
}
```

```java
class Driver {
    public static void displayDurationOutside(Movie obj) {
        System.out.println("Movie Duration = " + obj.getDuration());
    }

    Run | Debug
    public static void main(String[] args) {
        Movie avengers1 = new Movie(duration: 120);
        avengers1.display();

        Movie avengers2 = new Movie(duration: 150);
        avengers1.join(avengers2);
        avengers1.display();
    }
}
```

```
architaggarwal@Archits-MacBook-Air System Design % javac OOPS_Codes/08.ThisKeyword.java
architaggarwal@Archits-MacBook-Air System Design % java OOPS_Codes.Driver
Movie Duration = 120
Movie Duration = 270
architaggarwal@Archits-MacBook-Air System Design %
```

Q) Are there default parameterized constructors or functions in Java? What do you understand by Constructor Chaining?

```java
class Cuboid{
    int length;
    int breadth;
    int height;

    Cuboid(){
        // this.length = 1;
        // this.breadth = 1;
        // this.height = 1;
        this(1);
    }

    Cuboid(int side){
        // this.length = side;
        // this.breadth = side;
        // this.height = side;
        this(side, side, side);
    }

    Cuboid(int length, int breadth){
        // this.length = length;
        // this.breadth = breadth;
        // this.height = 1;
        this(length, breadth, 1);
    }

    Cuboid(int length, int breadth, int height){
        this.length = length;
        this.breadth = breadth;
        this.height = height;
    }
}
```

Constructor Chaining

One constructor calling another constructor using this() keyword!

```java
public static void main(String[] args){
    Cuboid obj1 = new Cuboid();
    Cuboid obj2 = new Cuboid(5);
    Cuboid obj3 = new Cuboid(5, 10);
    Cuboid obj4 = new Cuboid(5, 10, 15);

    System.out.println(obj1.length + " " + obj1.breadth + " " + obj1.height);
    System.out.println(obj2.length + " " + obj2.breadth + " " + obj2.height);
    System.out.println(obj3.length + " " + obj3.breadth + " " + obj3.height);
    System.out.println(obj4.length + " " + obj4.breadth + " " + obj4.height);

}
```

```
Finished in 109 ms
1 1 1
5 5 5
5 10 1
5 10 15
```

⇒ Constructor chaining call should be the first statement in the calling constructor.

```java
class Cuboid {
    int length, breadth, height;

    Cuboid() {
        this.length = 1;
        this.breadth = 1;
        this.height = 1;
    }

    Cuboid(int side) {
        this.length = side;
        this.breadth = side;
        this.height = side;
    }

    Cuboid(int length, int breadth) {
        this.length = length;
        this.breadth = breadth;
        this.height = 1;
    }

    Cuboid(int length, int breadth, int height) {
        this.length = length;
        this.breadth = breadth;
        this.height = height;
    }
}
```

```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Cuboid obj1 = new Cuboid();
        System.out.println(obj1.length + " " + obj1.breadth + " " + obj1.height);

        Cuboid obj2 = new Cuboid(side: 5);
        System.out.println(obj2.length + " " + obj2.breadth + " " + obj2.height);

        Cuboid obj3 = new Cuboid(length: 10, breadth: 15);
        System.out.println(obj3.length + " " + obj3.breadth + " " + obj3.height);

        Cuboid obj4 = new Cuboid(length: 10, breadth: 15, height: 20);
        System.out.println(obj4.length + " " + obj4.breadth + " " + obj4.height);

    }
}
```

Code without constructor chaining
Code redundancy!

```java
class Cuboid {
    int length, breadth, height;

    Cuboid() {
        this(side: 1);
    }

    Cuboid(int side) {
        this(side, side, side);
    }

    Cuboid(int length, int breadth) {
        this(length, breadth, height: 1);
    }

    Cuboid(int length, int breadth, int height) {
        this.length = length;
        this.breadth = breadth;
        this.height = height;
    }
}
```

```java
class Driver {
    Run | Debug
    public static void main(String[] args) {
        Cuboid obj1 = new Cuboid();
        Cuboid obj2 = new Cuboid(side: 5);
        Cuboid obj3 = new Cuboid(length: 5, breadth: 10);
        Cuboid obj4 = new Cuboid(length: 5, height: 10, breadth: 15);

        System.out.println(obj1.length + " " + obj1.breadth + " " + obj1.height);
        System.out.println(obj2.length + " " + obj2.breadth + " " + obj2.height);
        System.out.println(obj3.length + " " + obj3.breadth + " " + obj3.height);
        System.out.println(obj4.length + " " + obj4.breadth + " " + obj4.height);

    }
}
```

```
architaggarwal@Archits-MacBook-Air OOPS_Codes % javac 09.ConstructorChaining.java
architaggarwal@Archits-MacBook-Air OOPS_Codes % java Driver
1 1 1
5 5 5
5 1 10
5 15 10
```

Q) Is Java a PURE object oriented programming language? What are wrapper classes? What is autoboxing and unboxing?

Every entity should be class or object.
And it should follow all object oriented programming principles!

entity → predefined (primitive) or userdefined

int, boolean, char, long, float, double, short, byte

Answer: No

Solution ⇒ wrapper classes

int → Integer	float → Float
boolean → Boolean	double → Double
char → Character	short → Short
long → long	byte → Byte

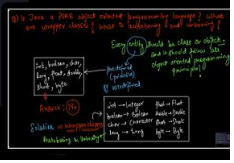
Autoboxing & Unboxing

```java
public static void main(String[] args) {
    // Primitive Stack Variable
    int a = 5;
    System.out.print(a + " ");

    // Integer Wrapper Class
    Integer aa = 10; // Autoboxing
    System.out.print(aa + " ");

    a = aa; // Unboxing
    System.out.print(a + " ");

    // Actual Object Creation, Getter and Setter
    Integer b = new Integer(value: 20);
    System.out.print(b.toString() + " ");

    a = b.intValue();
    System.out.print(a + " ");
}
```

```
architaggarwal@Archits-MacBook-Air OOPS_Codes % javac 10.WrapperClasses.java
10.WrapperClasses.java:17: warning: [removal] Integer(int) in Integer has been deprecated and marked for removal
        Integer b = new Integer(20);
                    ^
1 warning
architaggarwal@Archits-MacBook-Air OOPS_Codes % java Driver

5 10 10 20 20
```

```java
public static void functionsAndConst() {
    Integer a = 10;

    // Integer to String and String to Integer
    String b = a.toString();
    System.out.println(a + " Integer to String : " + b);

    String c = "256";
    Integer d = Integer.parseInt(c);
    System.out.println(c + " in Integer : " + d);

    // Various Number Conversions
    System.out.println(a + " Decimal to Binary : " + Integer.toBinaryString(a));
    System.out.println(a + " Decimal to Hexadecimal : " + Integer.toHexString(a));
    System.out.println(a + " Decimal to Octal : " + Integer.toOctalString(a));

    System.out.println("Integer MAXIMUM Range : " + Integer.MAX_VALUE);
    System.out.println("Integer MINIMUM Range : " + Integer.MIN_VALUE);

    // Various Other Functions
    System.out.println(a.compareTo(d));
    System.out.println(a.equals(d));
    System.out.println(Integer.max(a, d));
    System.out.println(Integer.min(a, d));
}
```

```
architaggarwal@Archits-MacBook-Air OOPS_Codes % java Driver
10 Integer to String : 10
256 in Integer : 256
10 Decimal to Binary : 1010
10 Decimal to Hexadecimal : a
10 Decimal to Octal : 12
Integer MAXIMUM Range : 2147483647
Integer MINIMUM Range : -2147483648
-1
false
256
10
```



```java
class MyInteger {
    private int data;

    public MyInteger(int data) {
        this.data = data;
    }

    public int getData() {
        return data;
    }

    public void setData(int data) {
        this.data = data;
    }

}
```

```java
public static void customWrapper() {
    MyInteger a = new MyInteger(data: 5);
    System.out.println(a.getData());

    a.setData(data: 10);
    System.out.println(a.getData());
}
```

```
architaggarwal@Archits-MacBook-Air OOPS_Codes % java Driver
5
10
```