# Java Basics

Class 5

**Class :-**

A class is collection object which is used to declare the data members(variables), member functions (methods) Constructor and block of statement(code).

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

- **Variables(data members)**
- **Methods(member functions)**
- **Constructors**
- **Blocks**

**Syntax:-** public class student{

contains variables, methods/function, blocks/statement

}

**Object:- An object is an instance of a class.** A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

syntax to create object of a class:-

Classname referenceVariable = new Classname();

//Object is used to call the non static members of class. We cannot call nonstatic members of class without creating the object of class

- Function/Methods :-

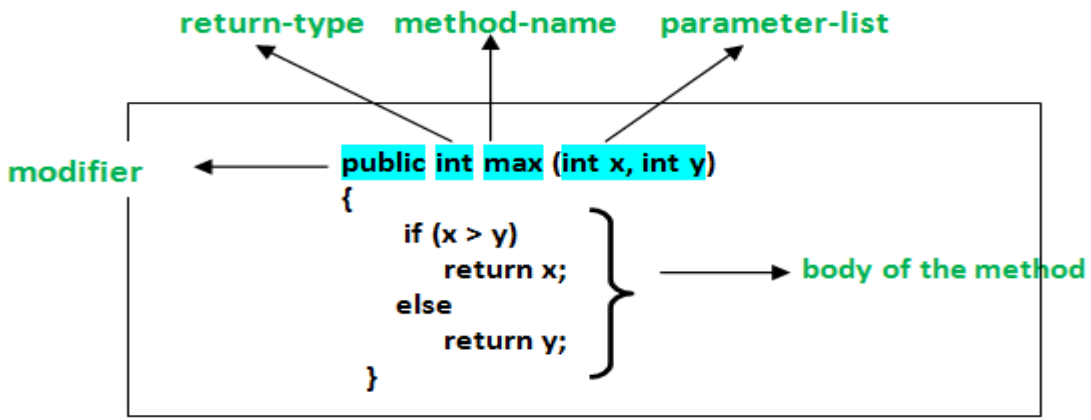
A method is a collection of statements that perform some specific task and return the result to the caller. Methods allow us to **reuse** the code without retyping the code.

Methods are always declared and defined outside the main method and inside the class.

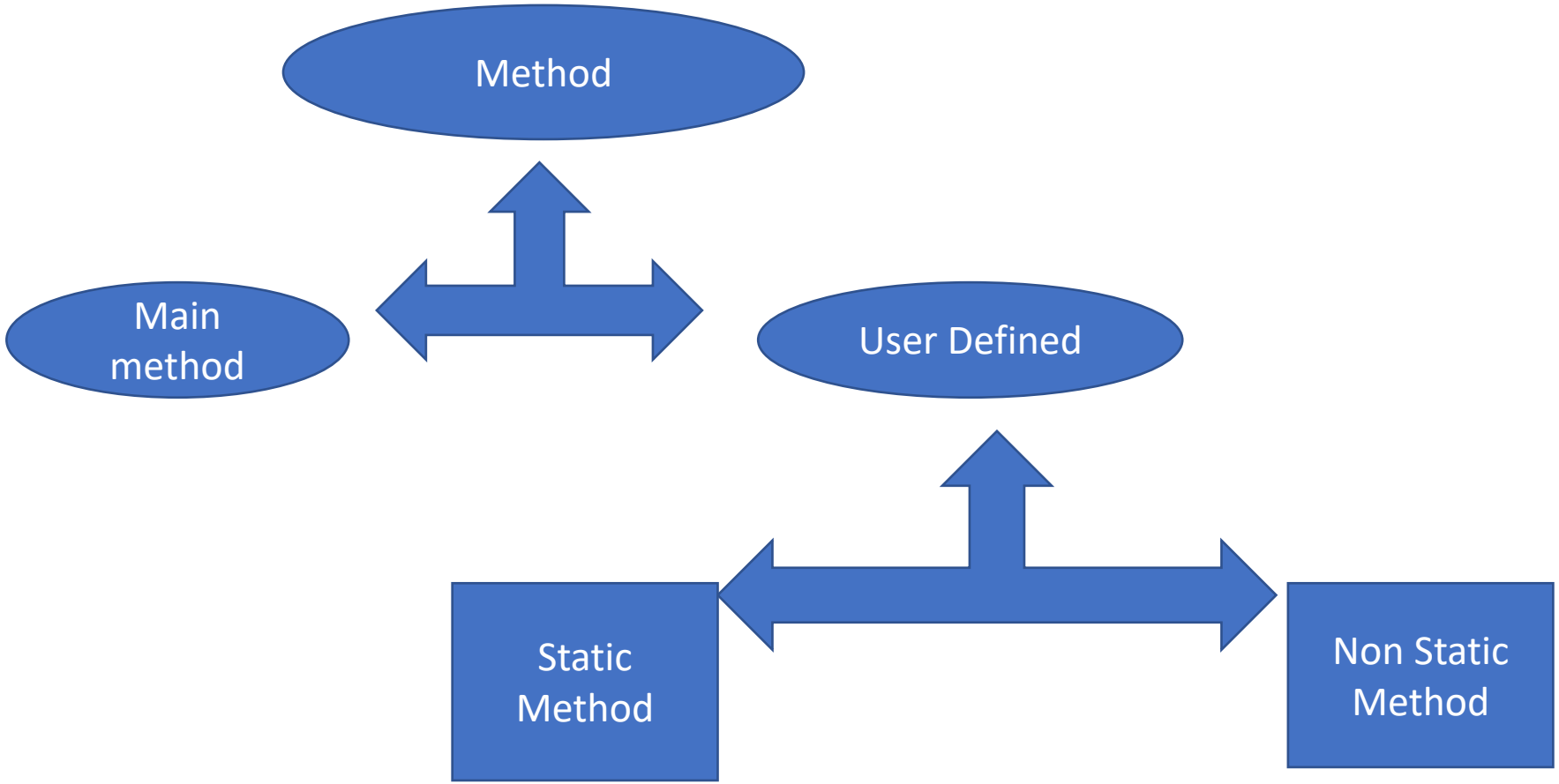
Some points to remember:-

- Methods are used to maintain the repeated code.
- Methods are always declared inside a class but outside main method.
- We cannot declare method inside a method.
- A method is always called inside the main method.

Syntax:-



Method Types:-



- **main method**:-

  Java execution always start with main method, Java main() method is always static, **so that compiler can call it without the creation of an object or before the creation of an object of the class**. In any Java program, the main() method is the starting point from where compiler starts program execution.

  Syntax :- public static void main (String args[])
  {

  }

- **static method** :- A method declared with static keyword is called as static method. A static method can be called directly using class name.

  Syntax :- static returnType method_name(arguments)
  {
  ----defination
  }

**Non static method :-** The method without static keyword, or it does not have any keyword.

syntax:-

```
return_type method_name(arguments)
{
        definition/code
}
```

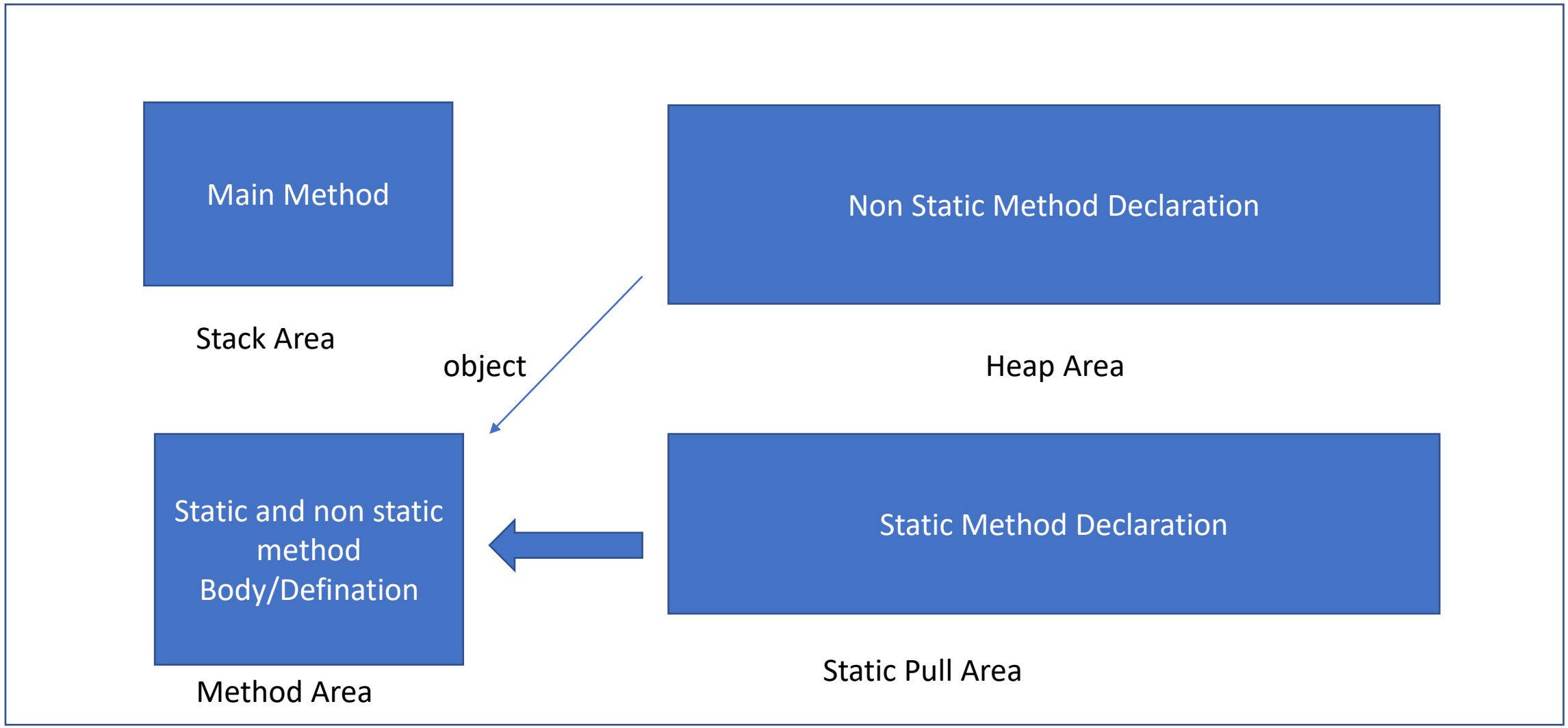
static keyword :- to call a static member directly we can call it with class name, no need of object creation

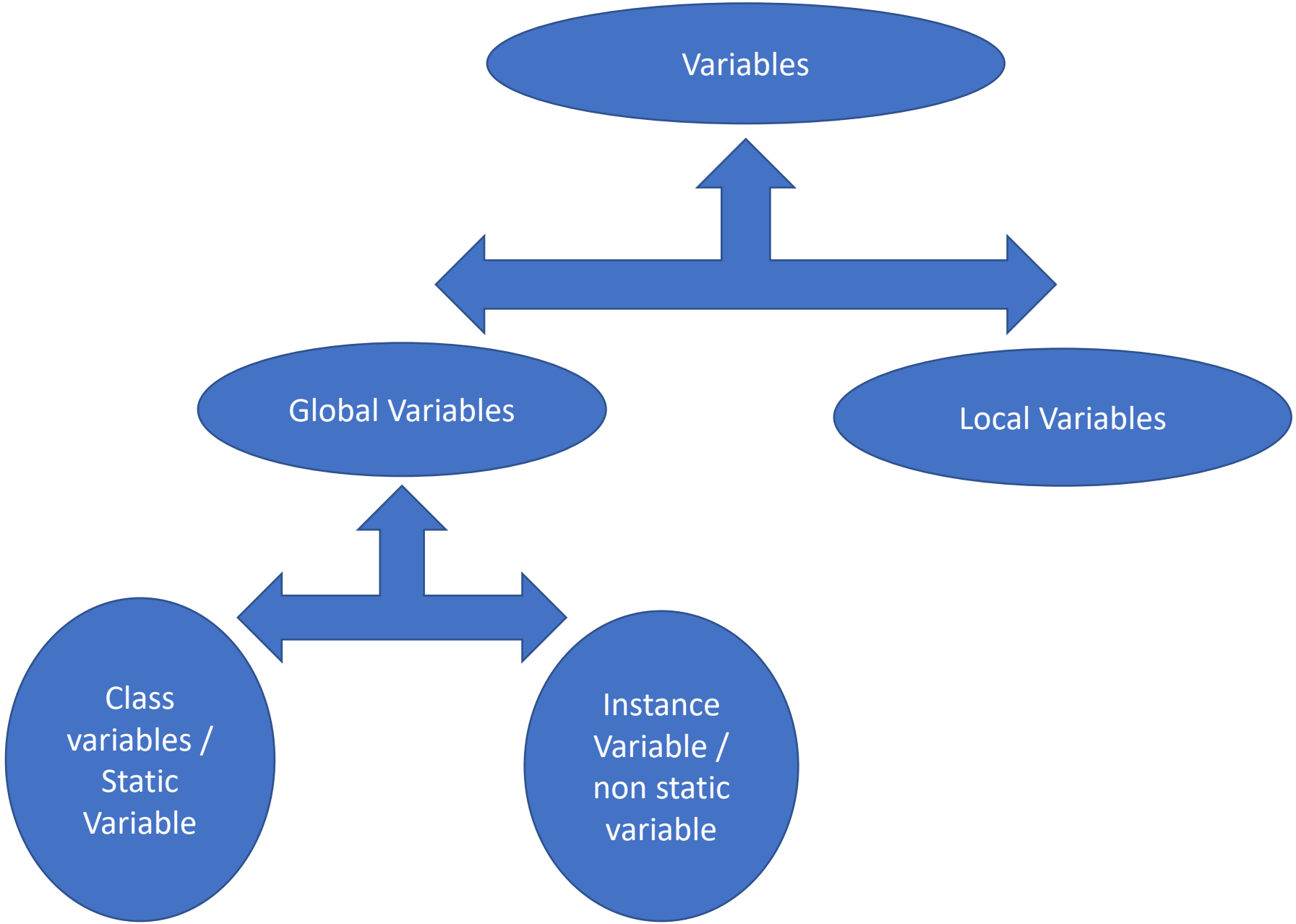
Loop.addition

To call a static member we directly use “classname.membername”

Non static :- Non static member’s doesnot have direct relation with class, so we need to create the object of class to call nonstaic members.

JVM Memory : - Java Virtual Machine Memory




Variables
Global Variables
Local Variables
Class variables / Static Variable
Instance Variable / non static variable

Global Variable:-

A variable declared inside the class but outside the method is known as Global variable, this variables can be accessed anywhere in the class.

Two Types :-

1. Static variables/Class Variables. (can be called in static as well as nonstatic methods)
2. Instance Variable/non static variable. (cannot be directly called in static method, we need to create object)

1. Static Variable/Class Variable:- A variable declared with static keyword is called as static variable or class variable, because to call this variable in other class we class name.

1. Instance variable/ non static variable:- A variable declared without static keyword is called as instance / non static variable, because in order to call this variable outside the class we have to create instance(object) of the class.
    1. A instance variable  cannot be called in static method.

Local variables :-

Local variables are declared locally inside a method, their access is restricted only inside the method. We cannot access a local variable outside the method.

```
public class Variables {
        static int a= 20;//Global Variable can accessed throughout the class

    public void addition() {
        int b =20;//Local variable cannot be accessed outside the method.
        System.out.println(b);

    }

    private void subtraction() {

        System.out.println(a);

    }

```

Access Specifier/Modifiers :-

Access Specifier are the keyword in java which defines the scope of the element within the project.

In Java we have 4 types of Access Specifier:-

- Public
- Default
- Protected
- Private

Public : -

Any element declared with public keyword , the scope of that element is throughout the project.

syntax:

```
public void test(){
}
```

Default :-

The scope of default access specifier is within the package. There is no keyword for default access specifier.

syntax:-

```
void test(){
}
```

Protected :-

The scope of the element declared with protected keyword has access within the package, but it can be accessed outside the package only in case of inheritance.

syntax:-

```
protected void test(){
}
```

Private :-

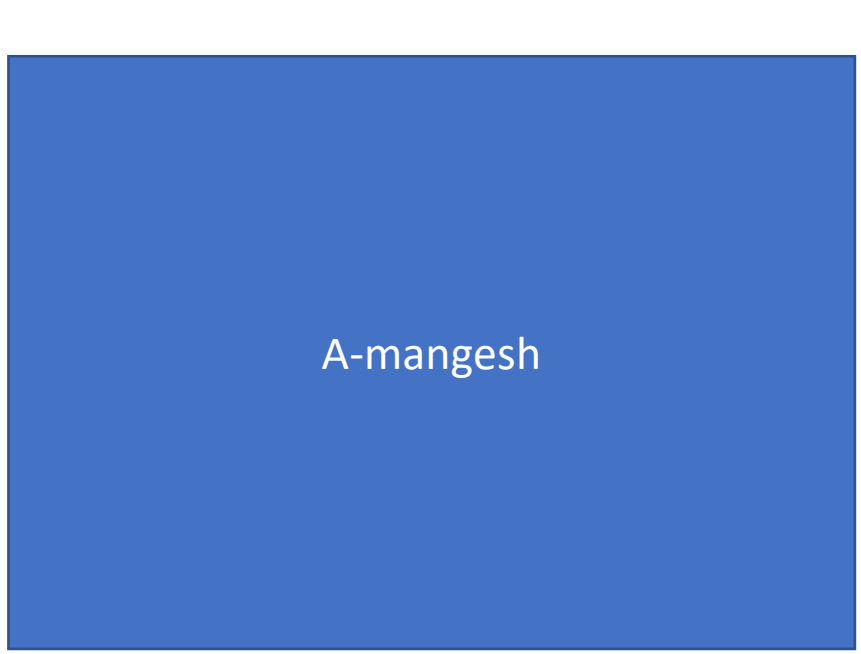
The scope of the element declared with private keyword has access within the class only.

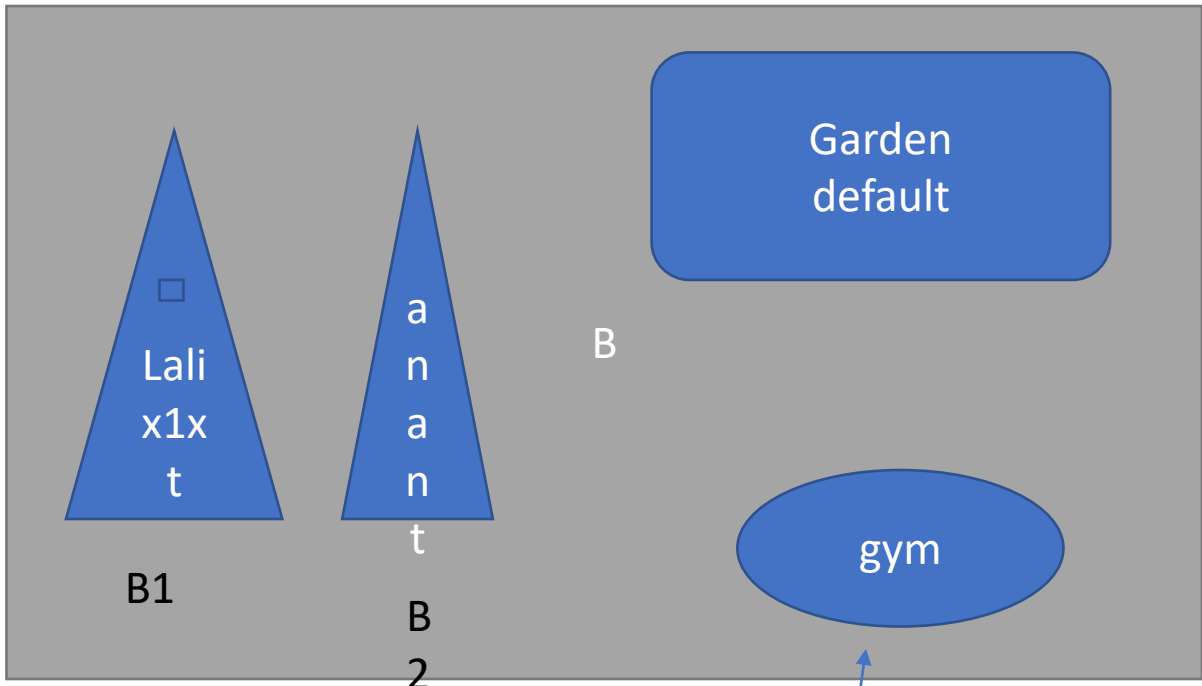
syntax:-

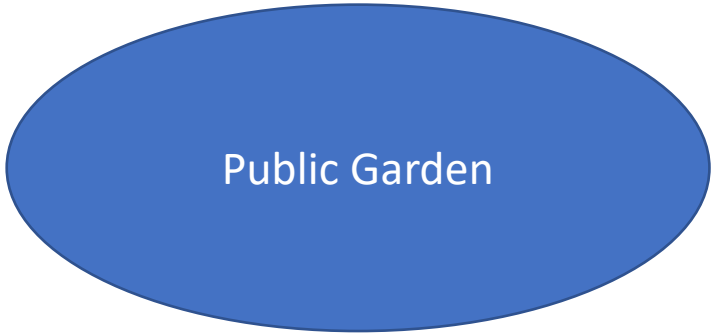
```
private void test() {
}
```

Points to remember :-

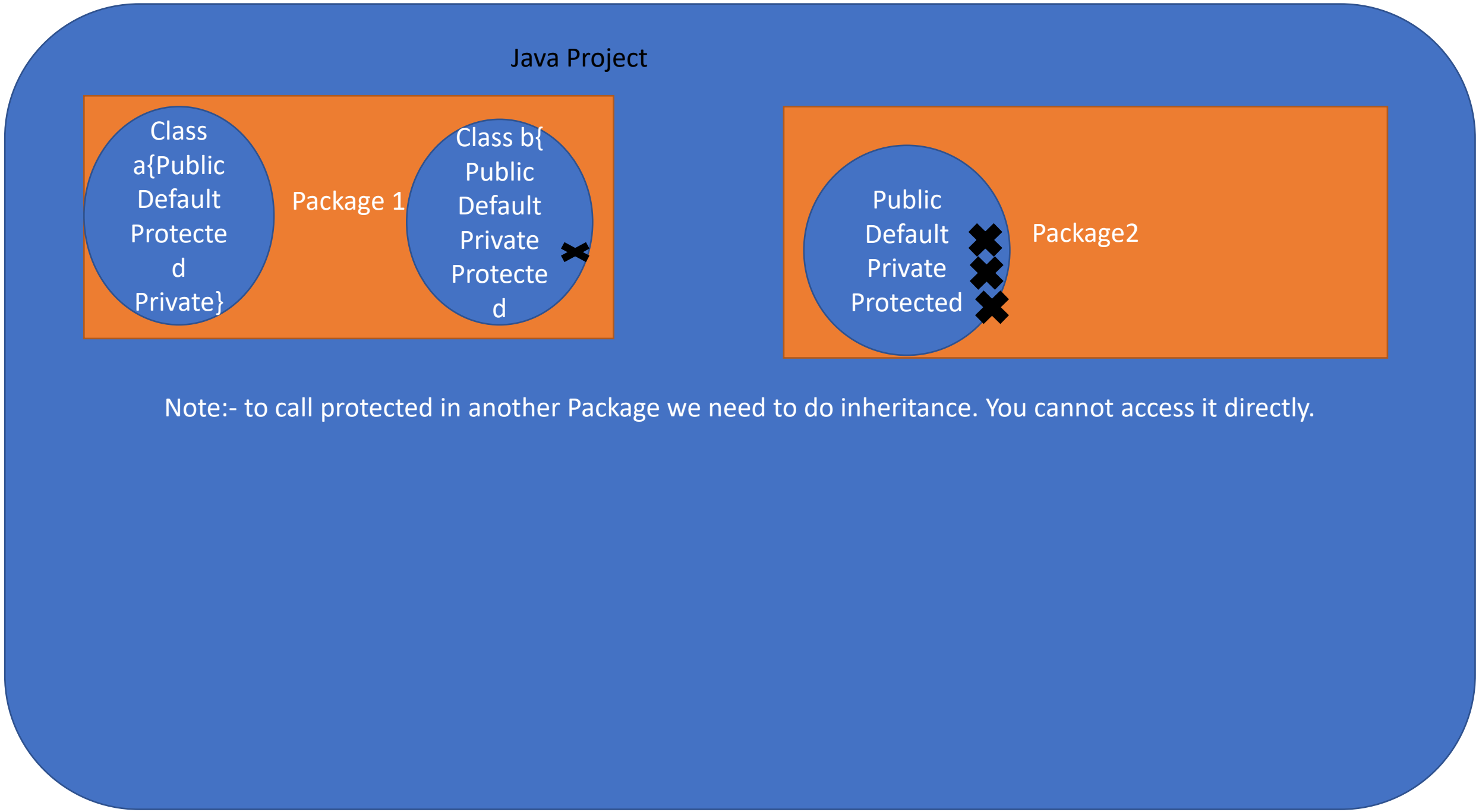
- A variable declared with private keyword cannot be accessed outside the class.
- A method declared with private keyword  cannot be called outside the class.




inheritance




Note:- to call protected in another Package we need to do inheritance. You cannot access it directly.

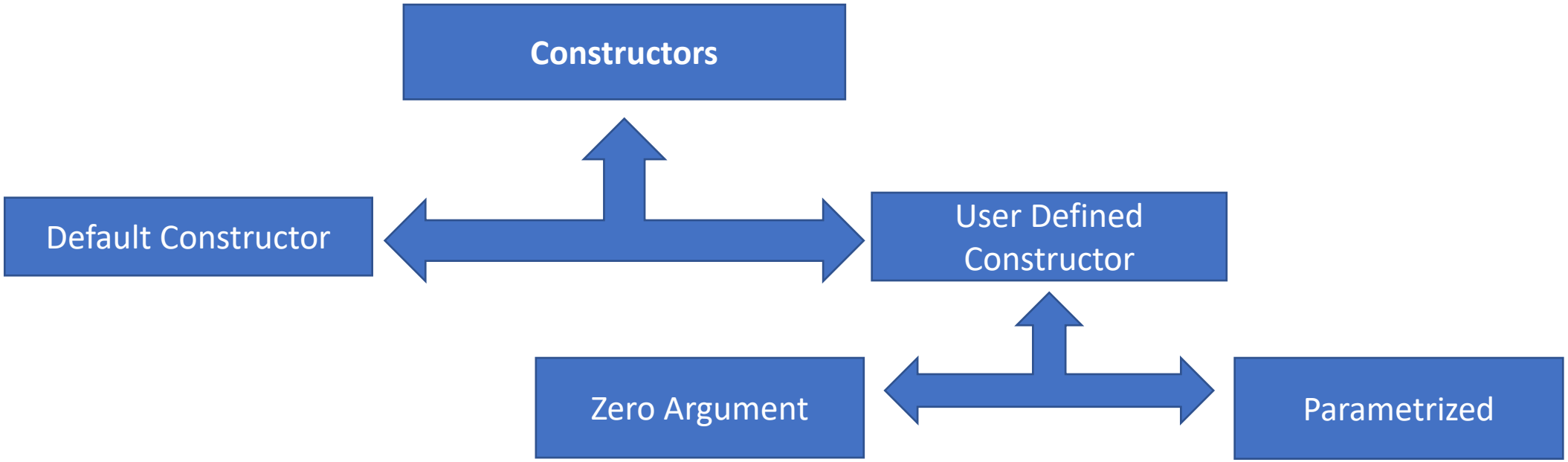
Constructor:-

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the Java memory.

Every time an object is created using the new() keyword, at least one constructor is called.

Constructor is a special member of class which is used to initialize the data members declared in same class and to load the non static elements (i.e. data members or member functions) of class into object or instance.

**Note:** It is called constructor because it constructs the values at the time of object creation. It is not necessary to write a constructor for a class. It is because java compiler creates a default constructor if your class doesn't have any.



| Java Constructor | Java Method |
| --- | --- |
| A Contructor is used to initialize a object | A method is used to expose the behavior of an object. |
| A constructor must not have a return type. | A method must have a return type. |
| The constructor is invoked implicitly. | The method is invoked explicitly. |
| The Java compiler provides a default constructor if you don't have any constructor in a class. | The method is not provided by the compiler in any case. |
| The constructor name must be same as the class name. | The method name may or may not be same as the class name. |

Points to remembers while creating a constructor:-

1.Constructor name must be the same as its class name
2.A Constructor must have no explicit return type
3.A Java constructor cannot be abstract, static, final.

- Default Constructor:- if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.

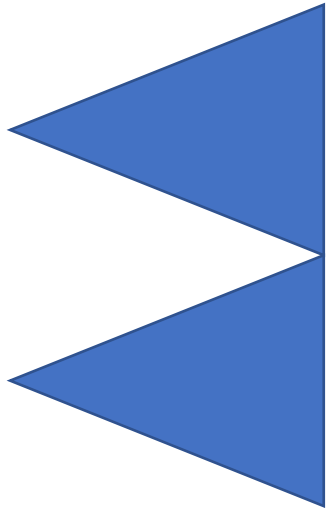
  The default constructor is used to provide the default values to the object like 0, null, etc., depending on the type.

Practice questions:-

1.Print Febonassi Series.

2.Print Pallendrom Number

3.Print
```
* * * * *
  * *   * *
  *       *
```



4. Find Smallest number and largest number for 4 numbers.

5 Print Prime Number .

6 Find LCM & HCF.

7 Armstrong Number

8 Factorial.