**Basic Concepts**

→ All the programs in Java are written inside a class.

**What is class?:**

1. It’s a design pattern/blue print/structure
2. Members of the class

* Methods/functions
* Variables

3. in Java “**class**” is a keyword

4. class declaration

Syntax:

<access\_specifier> <class\_type> **class** <class\_name> {

//Codes

}

**Keyword:**

→ It’s the reserved words

→ All keywords in Java are in lower case

**Variables:**

→ It’s a container/storage/memory location where we store some data

→ In a memory location we can keep/store different types of data.

Different types of data is called as Datatypes

**Datatypes:**

1. Primitive

2. Non-primitive/Derived

**Primitive:**

1. **boolean** → Always store **true**/**false** ----→ Default value: **false**

**boolean** b = false;

2. **char** → It stores single character -----→ Default value: **single space**

**char** c = ‘A’;

3. **byte** → Memory used: 1 byte

4. **short** → Memory used: 2 bytes Default value: **0**

5. **int** → Memory used: 4 bytes

6. **long** → Memory used: 8 bytes

7. **float** → Memory used: 4 bytes

8. **double** → Memory used: 8 bytes Default value: **0.0**

**Non-primitive:**

All class type datatypes are Non-primitive

Ex.: String, Arrays, Boolean, Character, Integer, Byte, Short, Long, Float, Double, etc

Default value: **null**

**Types of Variables:**

1. Local Variable

2. Instance/non-static variable

3. Static variable

Note: Generally combinely non-static and static variables are also called global variables

**Hierarchy of the java program**

class Main {

//Global variables

//Methods

public void test() {

//Local variables

//Business logic

}

}

**Local Variable:**

→ Declared inside the method

→ Can be accessed only inside the declared method

→ Can’t be accessed outside of the declared method. It still trying then will get compilation error

→ Mandatory to initialize the variable before it’s use else will get compilation error

**Global Variable:**

→ Declared outside of the method but inside the class

→ Can be used anywhere inside the class and also outside of the class\* (\* → based on the access modifier)

→ Not mandatory to initialize value to the variable. If initialized then it’ll consider the given value else it’ll be initialized with default value

**How to declare and initialize the valriable:**

1. <data\_type> <var\_name>; ---→ Declaration

Ex.: int i;

2. <data\_type> <var1>, <var2>, ...., <varN>;

Ex.: int i1, i2;

3. <var\_name> = value;

Ex.: int i; //Declared the variable

i = 10; //Initialized the variable

4. <data\_type> <var\_name>=value;

Ex.: int i=100;

**Rules for the identifier name:**

1. Identifier names only accepts a-z, A-Z, 0-9, \_ and $

2. Should not start with a number/digit

3. keywords can’t be used as a identifier name

**Methods:**

→ It’s a block developed with a set of/piece of codes performing a specific task

→ We are removing the code redundancy/code repetition

→ Code reusability

→ Readability increases

→ Maintainability

Note: All the methods are having 2 parts

A. Method signature

B. Method body

Ex.:

public void test() { ------→ signature

//body

}

If a method has the open and close curly braces then the method must have the body

**Types of method:**

1. Concrete: If a method has both signature and body

Ex.:

public void test() { ------→ signature

//body

}

2. Abstract: If a method has only the signature but no body

Ex.: public abstract void test1();

**Types of classes:**

1. Concrete: If a class contains only the concrete methods

2. Abstract:

→ If a class contains at least one abstract method

→ This class may contain both abstract and concrete methods

→ This class may contain all the abstract methods also

Note: The user of the class is basically called as Object

**Object:**

→ It’s a real time entity which has a state and behavior

Syntax to create an object:

<class\_name> <reference\_var\_name> = new <class\_name>(); → This is also called as instantiation

reference\_var\_name → Can be any name. It stores the address of the newly created object

new <class\_name>() → This is treated as the object

new → It’s a keyword used to create a new object upon every invocation

<class\_name>() → It’s the constructor used to construct the newly created object