class. ✓

object. ✓

variable/attributes/fields. ✓

Methods/ functions. ✓

Constructor ✓

modifier ✓

Geter & Seter (encapsulation)

Instance variable

overlapping. of methods.

access modifier < public
private ✓

premitive types

non-premitive/reference

static ✓

this

format.

Array (week 4)

Constructor

- Constructor name & class name are identical
- Constructor have no return type
- Constructor can be invoked using "new" keyword followed by class name & pare of brackets
- return statement can't be used in the body of the Constructor
- Constructor can be parameterized & also without arguments.

```
Public class Demo{
        int x;
        Demo(int y) {
                x = y;
        }

    Public static void main(String[] args){

        Demo obj = new Demo(5);
        system.out.Println(x);
        }

}
```

Mon Tue Wed Thu Fri Sat Date: / /20

- class is the template or blue print of object
- object is the instance of that class.
- Public class main{

```
Public class main{
            int x = 5 }

Public class mainTest{
      Public Static void main(String() args){
            main obj = new main();
 System.out.Println(obj.x)
```

- <u>class:</u>

A class is user defined blueprint or prototype from which object is created.

⇒ Modifier: A class can be Public or has default access

⇒ class keyword: class keyword is used to create a class.

⇒ class name: The name should begin with capital letter.

- object

it is a basic unit of object oriented programming & represent the real life entities. A typical java program creates many objects, which as you know, intract by invoking method.

variable/attribute/field.

* An attribute is another team for a field. it's typically a public constant or a public variable that can be accessed directly. ~~In this~~

```
    Public class main{
            int age;
            String name;
            .......}
```

age, names are attributes

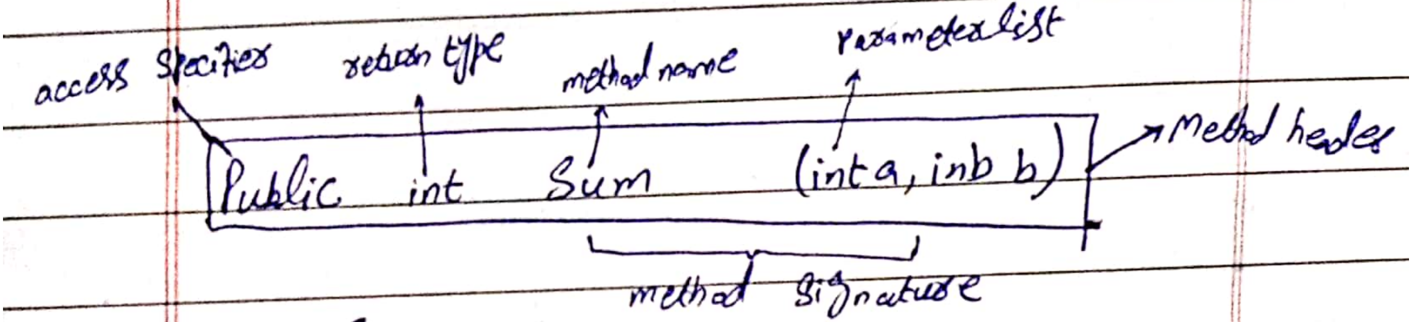
Methods:

A method is a way to perform some task. Simplify, the method in java is a Collection of instruction that perform a Specific task. it provides a ~~[illegible]~~ reusability of code. we can also easily modify code using methods.

⊛ method Declaration

The method decalaration provides information about method attributes, such as visiblity, return type, name & arguments.

it has Six Componets that are known as method header.



```
{
    //method body.
}
```

⇒ method Signature

Every method has a method Signature. it is a part of the method declaration. it ~~Consist~~ include the method name & parameter list.

⇒ Access Specifier/modifier.

Access specifier or modifier is the access type of the method. Java provides four type of access modifier

① Public ⇒ The method is accessible by all classes when we use public specifier in our application.

② Private ⇒ When we use a private access specifier, the method is accessible only the class in which it is defined.

③ Protected ⇒ when we use protected access modifer, the method is accessible within the Same package or Subclasses in a different package.

④ Default ⇒ when we do not use any access specifier in the method decalaration, java uses default access modifier by default. it is visible only from the Same package only

⇒ Return type.

Return type is a data type that the method returns. it may have primitive data type, object, Collection, void etc if the method does not return anything, we use void keyword.

⇒ method Name:

it is unique name that is used to define the name of a method. Suppose, if we are created a method for Substraction of two numbers, the method name must be Subtraction(). A method is invoked by its name.

⇒ Parameter list

it is the list of parameter Seperated by Comma & enclosed in a pair of Parentheses it ~~Cons~~ Contains the data type & variable name. if the method has no parameter, left the paranthesis blank.

⇒ Method body

it is a part of method decalation. it Consists all the ~~a~~ actions to be performed it is enclosed within a pair of curly braces.

Types of methods:

(1) Pre-defined method

(2) user-defined method.

① **Pre-defined method** ⇒ In Java Pre-defined methods are the method that is already defined in a java library is known as predefined methods. it is also known as Standard library method or built-in method. we can directly use these methods just by calling them in the program at any point Some pre-defined methods are length(), equals(), CompareTo(), sqrt() etc. —

② **user-defined method.**

The method written by the user or programmer is known as a user-defined method. These methods are modified according to the requirement

```
Public Static void findEvenOdd(int num)
{
      if(num%2==0)
      system.out.Println(num + "is even");
      else
      system.out.Println( num + "odd");
}
```

Static method:

A method that has Static keyword is known as Static method. In other words, a method that belongs to a class rather than an instance of class is known as a Static method. we can also create a Static method by using the keyword Static before the method name.

The main advantage of Static method is that we call it without creating an object

```
Public class Display
{
    Public Static void main(String[] args)
    {   Show();
    }

  Static void Show()
    {
        System.out.println("______");
    }

}
```

Instance method:-

The method of the class is known as instance method it is non-Static method defined in the class. before calling or invoking the instance method, it is necessary to create object of it's class.

Encapsulation.

Encapsulation in java is a process of wrapping code & data together into a Signal unit. for example a capsule which is mixed of Several medicines we can create a fully encapsulated class in java by making all the data members of a class private.

Setter & getter

```
Public class Person{
        Private String name;
    Public String getName(){
                return name;}
    Public void SetName(String newName){
            this.name=newName;}
```

- The get method return the value of variable name
- The set method takes a parameter(newName) and assigns it to the name variable. The this keyword is used to refer the current object.

Premetive types: int, byte, Short, long, float double, boolean & char

non-Premitive: String, array, class, object interface

Instance variable:-

instance variable in Java are non-static variables which are defined in a class outside any method, Constructor or a block Each instantiated object of a class has a separate copy or instance of that variable. An instance of variable belongs to a class.