**Selenium:**

It is a set of tools and libraries that automates web browser action

In most simple words, selenium provides tools that can interact with browser and can automate browser actions like click, input, select, navigate etc. with the help of scripts

**What is Selenium webdriver:**

Webdriver is one of the components in selenium. Webdriver is a java interface. Webdriver is an API(Application Programming interface)

**Different Types of Selenium Locators:**

There are several types of locators in Selenium that you can use to locate web elements on a web page. The most commonly used locators are:

1. **ID Locator:** It locates a web element based on its ID attribute. This locator is generally considered the fastest and most reliable, as ID attributes are supposed to be unique on a web page.
2. **Class Locator**: It finds a web element based on its class attribute. This locator is helpful when multiple elements on a web page share the same class attribute.
3. **Name Locator**: It helps in locating a web element based on its name attribute. This locator is helpful when interacting with form elements like text boxes, radio buttons, and checkboxes.
4. **XPath Locator**: It searches and navigates to the web element based on its position within the DOM tree or its attribute values. XPath locators are robust but can be slower and more brittle than other locator types.
5. **CSS Locator**: The CSS locator is used to locate web elements based on their CSS selector. The CSS selector is a pattern that is used to select web elements based on their attributes, such as ID, class, and name.
6. **Tag Name Locator**: Based on its HTML tag name, elements are located. This locator is helpful when you must interact with all elements of a specific type, such as all links or all buttons.
7. **Link Text Locator**: Based on the text of a hyperlink, a link text locator finds a web element in the DOM. This locator is helpful when interacting with a specific link on a web page.
8. **Partial Link Text Locator**: Uses a partial match of the text of a hyperlink to find a web element in the DOM. This locator is helpful when interacting with a link with a changing or dynamic text value.

To use a Selenium locator, you first need to create an instance of the WebDriver class, representing the browser you want to automate. Once you have a WebDriver instance, you can use the findElement() or findElements() method to locate web elements using the locator of your choice.

|  |  |
| --- | --- |
| **Locator Type** | **Method Name** |
| ID | find\_element\_by\_id |
| Name | find\_element\_by\_name |
| Class Name | find\_element\_by\_class\_name |
| Tag Name | find\_element\_by\_tag\_name |
| Link Text | find\_element\_by\_link\_text |
| Partial Link Text | find\_element\_by\_partial\_link\_text |
| CSS Selector | find\_element\_by\_css\_selector |
| XPath | find\_element\_by\_xpath |

**Methods:**

FindElement: Will return Single Element

FindElements: TagName, Class🡪TagName and class can be used to locate group of elements

**Locators:** We can identify various Elements on Web by using Locators. Locators are addresses that identify a web element uniquely within the page

**By ClassName:**

List<WebElement>headerLinks=driver.findElements(By.className("");

System.out.println("total number of header links:"+headerLinks.size());

**By TagName:**

List<WebElement>links=driver.findElements(By.tagName("");

System.out.println("total number of links:"+links.size());

Webdriver driver=new Chromedriver();

**Locators Syntax:**

Syntax to use locators in FindElement method

**By id:**

driver.findElement(By.id(""))

**By Name:**

driver.findElement(By.name("")).sendkeys("");

**For Boolean:**

boolean.logoDisplaystatus=driver.findElement(By.id()).isDisplayed();

**By LinkText:**

driver.findElement(By.linkText("")).click();

**By PartialLinkText:**

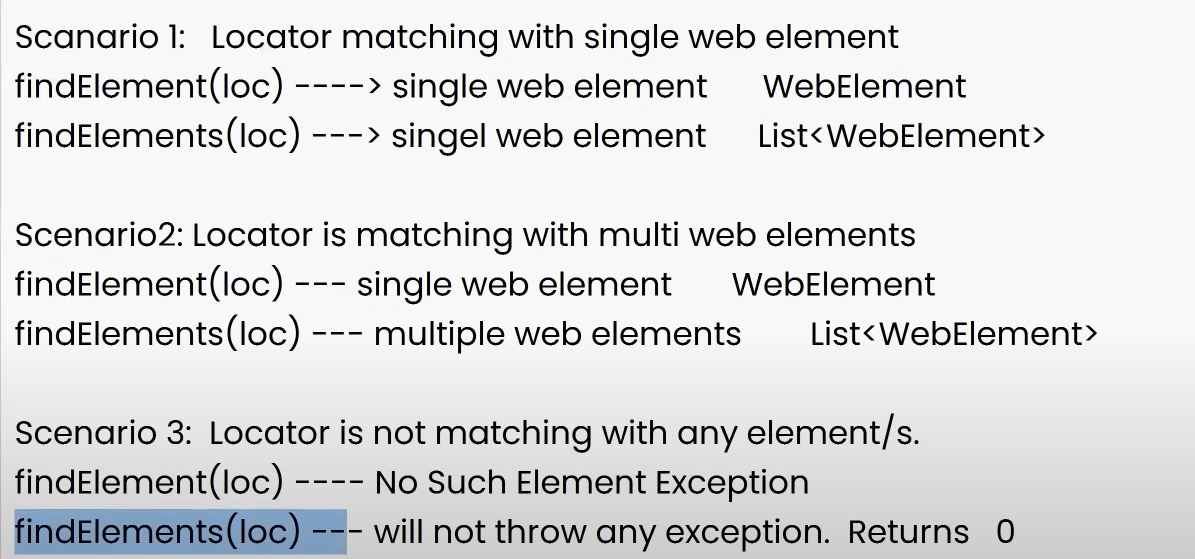
driver.findElement(By.partialLinkText().click();

**Notes:**

TagName and class can be used to locate group of elements

isDisplayed() will return boolean value

hred represents the target



**Variables:**

Variable is a container which can hold the data. To represent data we need variable

**Data Types:**

Represents the Type of Data

**Premitive** -Can store single value

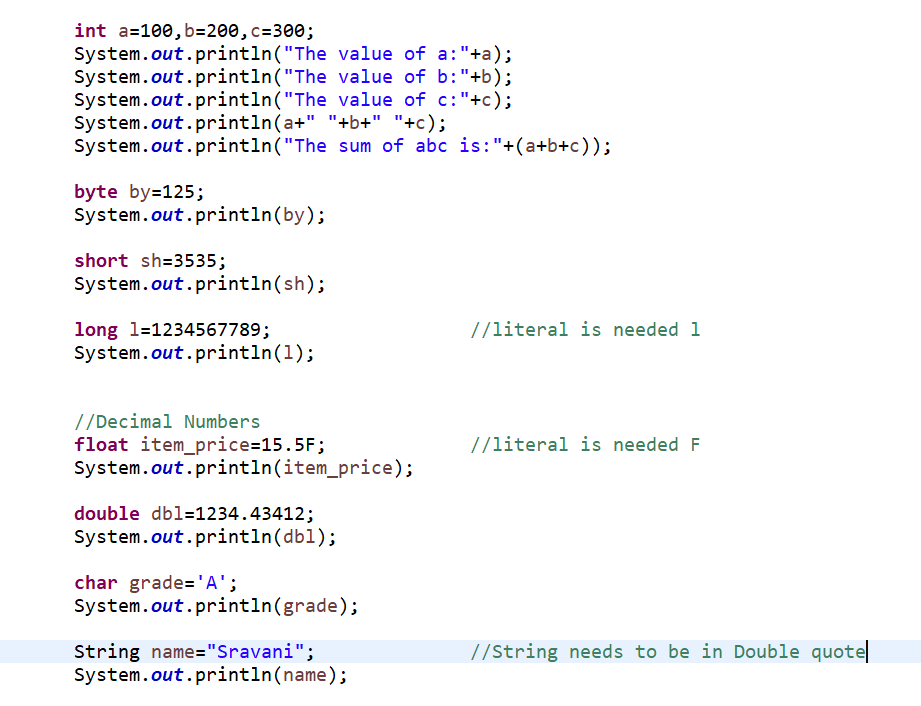
Byte, short, int, long----->number without decimal

float, double------->decimal number

char-------->single character

boolean------>true, false

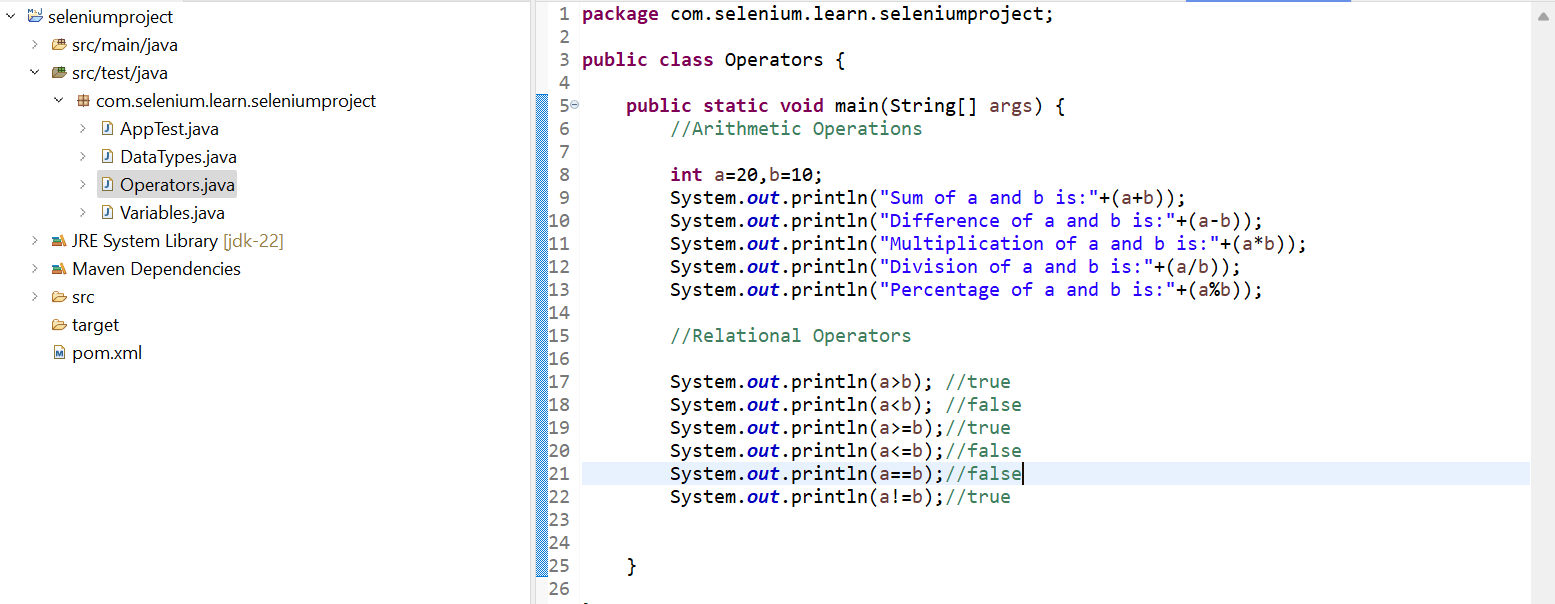
|  |  |  |
| --- | --- | --- |
| DataType | Size | Description |
| byte | 1 byte | Stores whole numbers from -128 to -127 |
| short | 2 bytes | Stores whole numbers from -32,768 to 32,767 |
| int | 4 bytes | Stores whole numbers from -2,147,483,648 to 2,147,483,647 |
| long | 8 bytes | Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 |
| Float | 4 bytes | Stores fractional numbers. Sufficient for storing 7 decimal digits |
| double | 8 bytes | Stores fractional numbers. Sufficient for storing 15 decimal digits |
| boolean | 1 bit | Stores true or False values |
| char | 2 bytes | Stores a Single character/letter |

2)Non-Premitive-Can store Multiple values String, ArrayList, HashMap, HashSet

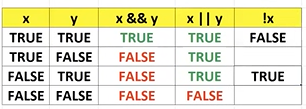
**Operators:**

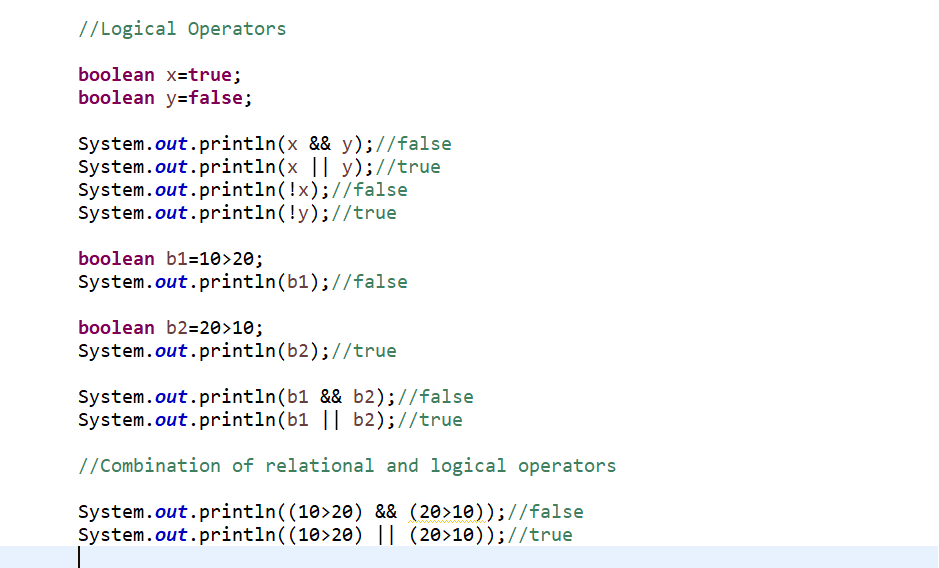
**Arithmetic Operators**: +,-,\*,/,%

**Relational Operators**: >,>=,<,<=,==,!=(Returns Boolean Values)

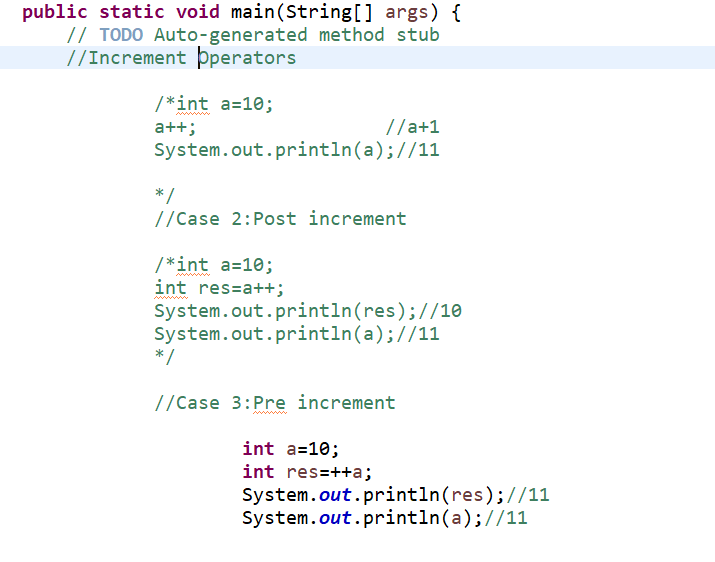


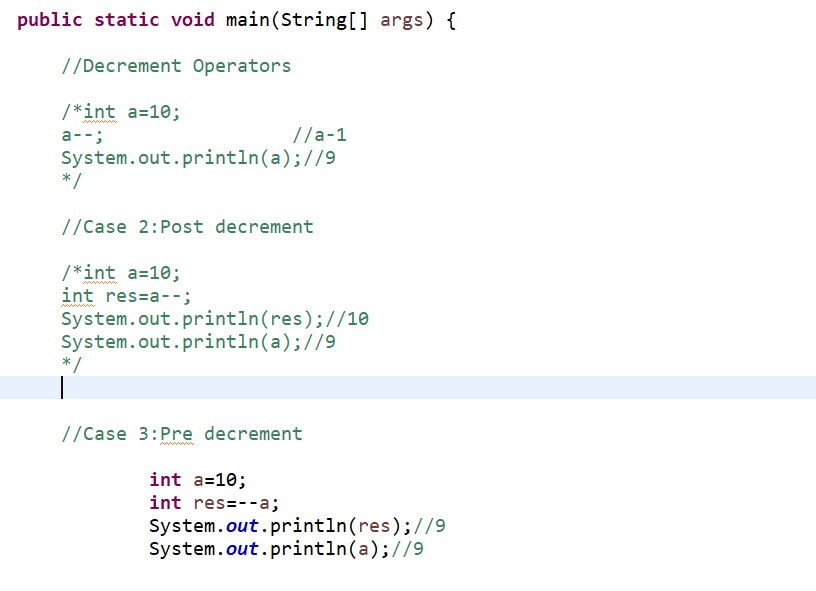
**Logical Operators**: &&(And),||(Or),!(Not)--->Returns Boolean Values--->Works between 2 boolean values



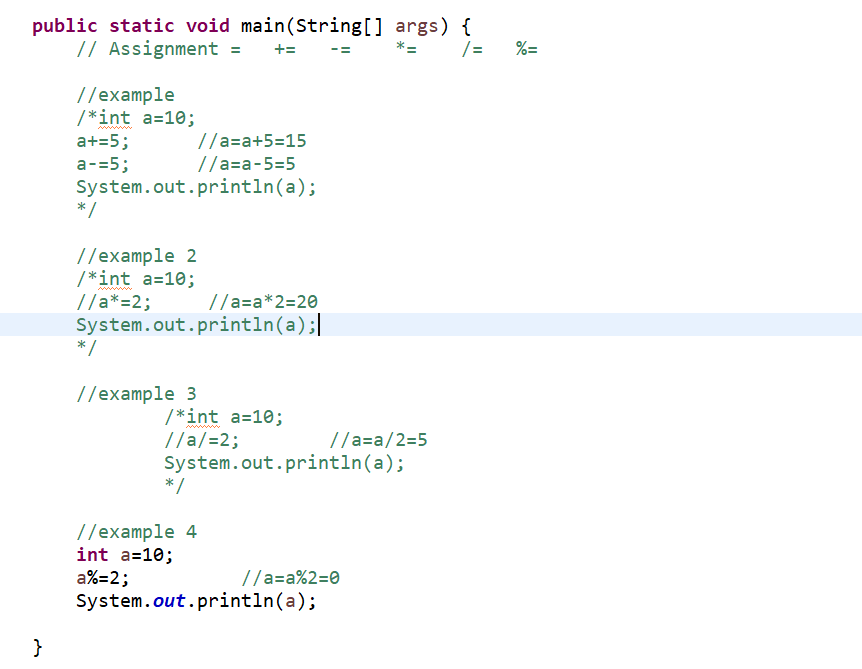


**Increment, Decrement Operators**:++,--

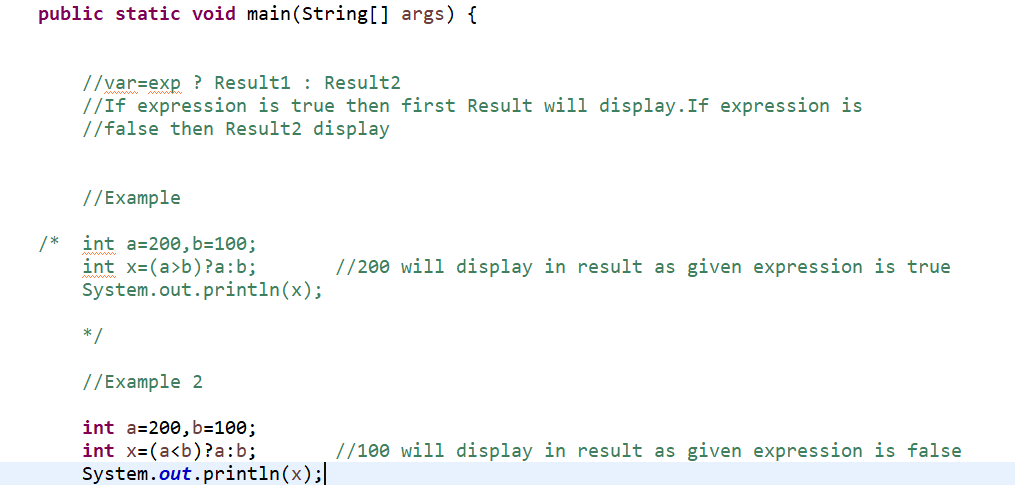




**Assignment Operators:** = += -= \*= /= %=



**Conditional or Ternary Operator:**  ?:



How to swap to numbers?

**Control Statements:**

1)Conditional Statements

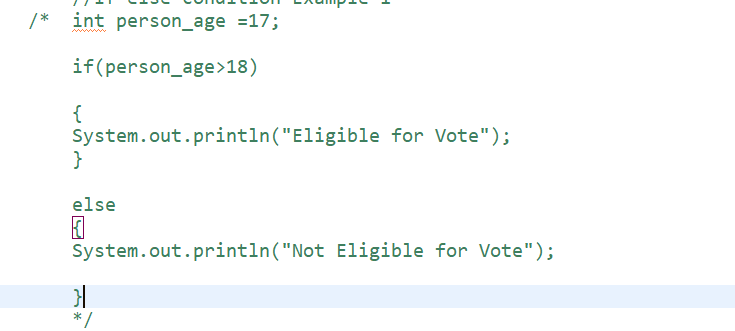
2)Looping/iterative statements

3)Jumping Statements

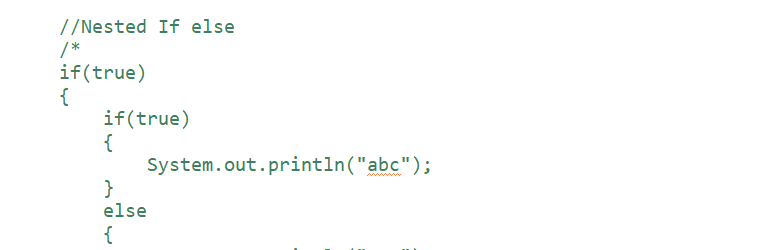
**Conditional Statements:**

1)if

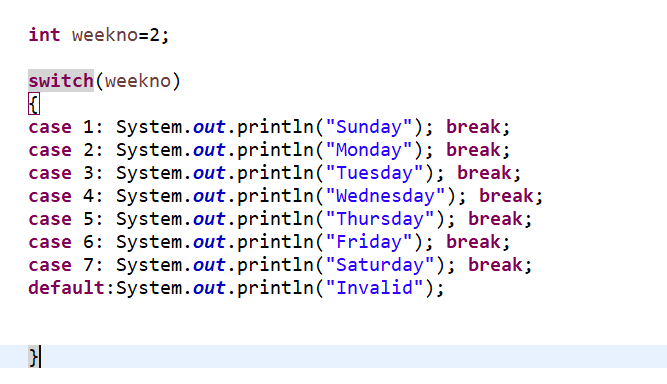
**2)if else**



**3) nested if else:**



**4) switch case:**



**TYPES OF LOOPS:**

* 1)While Loop
* 2)Do while Loop
* 3)For Loop
* 4)Enhanced for Loop

Three things we need to concentrate while using Looping statement

* 1)Initializing(where to start)
* 2)Condition(how many times)
* 3)Increment/Decrement

**While Loop Structure:**

Initialization;

While(condition)

{

Statements;

Inc/dec;

}

Example1:print 1………………10 numbers

int i=1;

while (i<=10)

{

System.out.println(i);

i++;

}

**do..while loop:**

do

{

statements;

inc/dec

}while(condition);

Example: Print 1……………………………..10

int i=1;

do

{

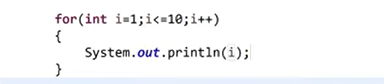
System.out.println(i);

i++;

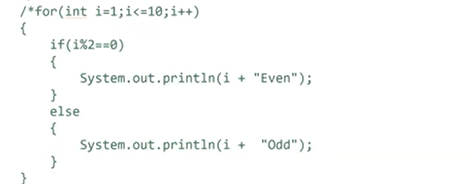
}while(i<=10);

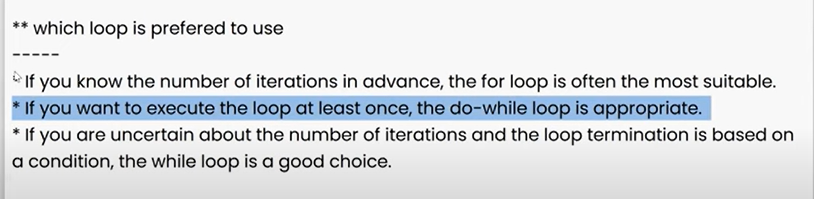
**For Loop:**

for(initialization;condition;inc/dec)



Example to print Even and odd numbers by using For loop:





**Jumping Statements:**

* 1)Break
* 2)Continue

**Control statements:**

* 1)Conditional statements------if, if-else, nested if else, switchcase
* 2)looping/iterative statements------while, do-while, for loop
* 3)jumping statements------break, continue

**Arrays:**

Array is a collection of elements of same data type. We can store multiple values into a single variable

Single Dimensional Array Operations:

1)Declare an array

2)Add values into array

3)Find size of an array

4)Read single value from an array

5)Read multiple vales from an array

**Declare an array:**

Example 1:

int a[]=new int[5];

a[0]=100;

a[1]=200;

a[2]=300;

a[3]=400;

a[4]=500;

Example 2:

int a[]={100,200,300,400,500};

**Find length of an array:**

System.out.println(a.length); //output 5

**Read single value from an array:**

System.out.println(a[4]); //output 500

Reading all the values from array

**//Normal for loop**

for(int i=0;i<a.length;i++) or for(int i=0;i<=a.length-1;i++) //i<=4 i<5 i<=a.length-1 i<a.length

{

System.out.println(a[i]); //100 200 300 400 500

}

**//enhanced for loop**

for(int x:a)

{

System.out.println(x); //100 200 300 400 500

}

**Object Oriented Programming:**

**Object Oriented Programming concepts:**

* Class
* Object
* Polymorphism
* Encapsulation
* Inheritance
* Data Abstraction

**Class**: Class is collection of attributes and behaviour. Class is a collection of variables and methods

Group of objects are called as a class. In the below Animal, students, Employee are classes

Animal--🡪Dog, Elephant, horse etc

Students-🡪Kim, David, Scott etc

Employee-🡪John, smith etc

1) Class is collection of variables and methods

2) Class is logical entity/blue print

3) Will not occupy space in the memory

**Object:**

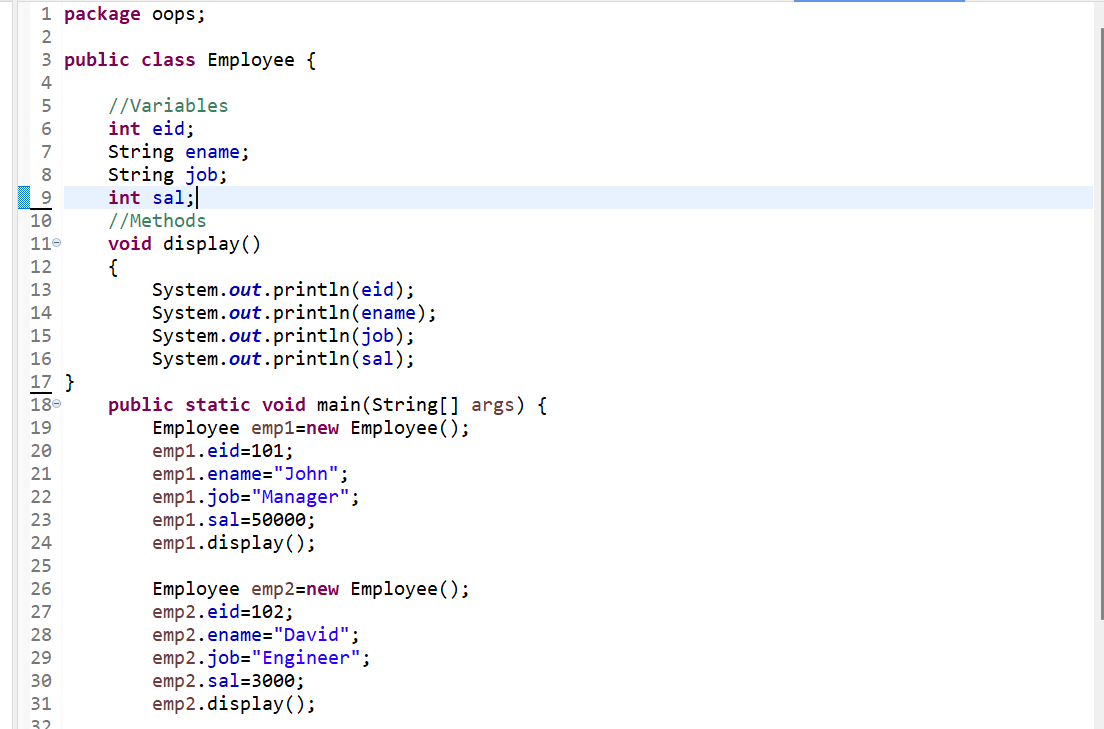
1) Object is an instance of a class/object will be created by using class

2) Object is physical entity

3) Occupy space in memory

4) We can create multiple objects for single class

Example: Employee emp1=new Employee(); //where emp1 is the object



**Methods:**

Block or group of statements which will perform certain task

We have to call the method through object

1)No params No return value

2)No params Return value

3)Takes parameter No return value

4)Takes Parameter Returns value

String s=new String(“Welcome”);

Int len=s.length()----------------------no parameter, returns value

1) If the method is returning some value we need to capture that return value in a variable

**Polymorphism:**

Polymorphism allows us to perform a single action in different ways. In other words, polymorphism allows you to define one interface and have multiple implementations.

**Types of Java Polymorphism:**

In Java Polymorphism is mainly divided into two types:

Compile-time Polymorphism

Runtime Polymorphism

**Method Overloading:**

We can achieve Polymorphism using overloading.4 rules are applicable for overloading

1)Methods name should be same

2)Number of parameters should be different

3)Data types of parameters should be different

4)Order of parameters should be different

**Constructor overloading:**

Note: With one object we can invoke only one constructor

Constructor is used for only assignment

Constructor overloading can also be achieved by 4 rules

1)Number of parameters should be different

2)Data types of parameters should be different

3)Order of parameters should be different

**Encapsulation:**

Wrapping of data and data into single unit(class)

1)All variables should be private

2)For every variable there should be 2 methods(get & set)

3)Variables can be operated only through the methods