Day 7 - 06th June 2025

Java

| Java basics | Data Types, Variables, Operators, Control Statements, Loops, Arrays, Classes, Objects, |
| --- | --- |

Open labs

Then check for Java version

In cmd type

Java –version

or

Java -v

17 installed

Plz check for any of the ide’s is installed

Eclipse

Or intellij

Or vs code

Variable

X, y, name, age,

int —> data type

int age; ====> variable declaration

char name;

Assign value to a variable

age = 10;

name = ‘k’;

Declaring and assigning value to a variable

int age = 20;

char name =’M’;

Types of data types👍

Primitive

Int, char, float…

and non primitive

String, array…

Methods:

Public ===> anyone can access… ===> access modifiers in java (access specifiers in c++)

main ====> boss of your program ===> entry and exit point of the program

static ===> fixed ===> retains its value.. Variables X Static

Void ===> return type ====> return 0 value ===> success

Or

int add() {

int x =20;

return x;

}

public static void main(String[] args){

//body of the main

}

===============================

Try in c

================================

int add(){

int x = 10;

static int y = 20;

x++;

y++;

printf(“%d”, x);

printf(“%d”, y);

}

main(){

add(); ===>

add();

add();

}

—----$$$$$$$$$$$$$$$$$$------------------

#include <stdio.h>

int add()

{ int x=10;

static int y=20;

x++;

y++;

printf("%d\n",x);

printf("%d\n",y);

}

void main() {

add();

add();

add();

}

======================================================

Task001:

Wap to display greetings

public class Greetings {

public static void main(String[] args) {

System.out.println("Hello! Welcome to Java Programming!");

}

}

3 min 10.35 to 10.38

Task002:

Wap to create a add method and call the method 3 times ..

Hint in method add declare variables and display them

public class AddMethod {

static void add() {

int x = 5;

int y = 10;

System.out.println("Sum is: " + (x + y));

}

public static void main(String[] args) {

add();

add();

add();

}

}

5 min 10.39 to 10.43

Task003

Write a Program in Java to Add two Numbers.

Input: 2 3

Output: 5

public class AddNumbers {

public static void main(String[] args) {

int num1 = 2;

int num2 = 3;

int sum = num1 + num2;

System.out.println(sum);

}

}

10.44 to 10.49

Task004

Write a Program to Swap Two Numbers

Input: a=2 b=5

Output: a=5 b=2

10.50 to 10.55

Task005

Create a code in which you have 4 methods add, subtract, multiply and divide (return type int) with a main [method..to](http://method..to) call all the other methods

public class Calculator {

static int add(int a, int b) {

return a + b;

}

static int subtract(int a, int b) {

return a - b;

}

static int multiply(int a, int b) {

return a \* b;

}

static int divide(int a, int b) {

return a / b;

}

public static void main(String[] args) {

System.out.println("Main started");

System.out.println("Sum of 2 numbers is " + add(10, 5));

System.out.println("Diff of 2 numbers is " + subtract(10, 5));

System.out.println("Product of 2 numbers is " + multiply(10, 5));

System.out.println("Division of 2 numbers is " + divide(10, 5));

System.out.println("Main ended");

}

}

Out put:

Main started

Sum of 2 numbers is …..

Diff of 2 numbers is —-

Product of 2 numbers ….

Division of 2 numbers is ….

Main ended

10.56 to 11:06

Operators 👍

Types of operators:

Arithmetic operators : +, -,\*,/, % (% modulo gives you remainder)

Relational Operators : > , < , <=,>=

Logical operators : &&, ||, !, !=

Comparison operators: ==

Assignment operators : =

Unary operators = ++, - - (uses single operand)

Binary operators : same as Arithmetic etc..uses 2 operands

Ternary operators : uses 3 operands

Eg: (a>b)?“True”:“false”;

Task006

Write a program to check if a is greater or b.. Use ternary op

public class CompareNumbers {

public static void main(String[] args) {

int a = 10;

int b = 5;

String result = (a > b) ? "a is greater" : "b is greater";

System.out.println(result);

}

}

11.07 to 11.12

Task007

Write a program to take input from the user and display it to the user

import java.util.Scanner;

public class UserInput {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your ID:");

String id = sc.nextLine();

System.out.println("Enter your Password:");

String pwd = sc.nextLine();

System.out.println("Hi,");

System.out.println("\tYour login id is " + id);

System.out.println("And your pwd is " + pwd);

sc.close();

}

}

Input:

Id : Prasunamba

Pwd: 123456789

Output:

Hi ,

Your login id is Prasunamba

And your pwd is \*\*\*\*\*\*\*\*\*

HInt :

For scanner … import java.util.scanner;

Scanner sc = new Scanner([System.in](http://system.in));

Id = sc.nexLine();

11.13 to 11.20

Task008

Write a program to create a class named Customer

Call the customer class in Task008 class using an object.

Hint

In the main method

Class Customer{

void accept(){

sysout(“accept customer called”);

}

Void display(){

sysout(“display customer called”);

}

}

Public class Test008{

psvm(String[] args){

Customer cobj = new Customer();

cobj.accept();

cobj.display();

}

}

Ans:

class Customer {

void accept() {

System.out.println("accept customer called");

}

void display() {

System.out.println("display customer called");

}

}

public class Task008 {

public static void main(String[] args) {

Customer cobj = new Customer();

cobj.accept();

cobj.display();

}

}

11.21 to 11.31

Task009:

Wap to check the greater of 2 numbers

Hint:

Use if else

If ( num1 > num2){

sout(“num1 is greater”);

}

Else {

sout(“num2 is greater”);

}

Ans: public class CompareTwo {

public static void main(String[] args) {

int num1 = 15;

int num2 = 10;

if (num1 > num2) {

System.out.println("num1 is greater");

} else {

System.out.println("num2 is greater");

}

}

}

11.32 to 11.37

task 010

Wap to check greater of 3 numbers

Hint 👍

Use elseif

public class CompareThree {

public static void main(String[] args) {

int num1 = 15;

int num2 = 10;

int num3 = 20;

if (num1 > num2 && num1 > num3) {

System.out.println("num1 is greatest");

} else if (num2 > num1 && num2 > num3) {

System.out.println("num2 is greatest");

} else {

System.out.println("num3 is greatest");

}

}

}

11.38 to 11.43

Task11:

Wap to check if week days

1 ===> sunday

2 ===> monday

So on

8 and above ===> invalid input

Hint : use Switch case

public class WeekDays {

public static void main(String[] args) {

int day = 3;

switch(day) {

case 1:

System.out.println("Sunday");

break;

case 2:

System.out.println("Monday");

break;

case 3:

System.out.println("Tuesday");

break;

case 4:

System.out.println("Wednesday");

break;

case 5:

System.out.println("Thursday");

break;

case 6:

System.out.println("Friday");

break;

case 7:

System.out.println("Saturday");

break;

default:

System.out.println("Invalid input");

}

}

}

Task 012

Wap to check loginid and password validation

Hint use while loop

Scanner sc = new Scanner(System.in);

String loginid = “Prasunamba”

String pwd = “12345867”

Int Count = 0;

While (loginid == “Prasunamba” && pwd == “12345867”){

sout(“ you have logged in for ”+ count++ +” times”);

sout(“enter ur login id and password”);

loginid = sc.NextLine();

pwd = sc.NextLine();

}

Do while also use 👍

Scanner sc = new Scanner(System.in);

String loginid = “Prasunamba”

String pwd = “12345867”

Int Count = 0;

do{

sout(“ you have logged in for ”+ count++ +” times”);

sout(“enter ur login id and password”);

loginid = sc.NextLine();

pwd = sc.NextLine();

}While (loginid == “Prasunamba” && pwd == “12345867”);

sc.close();

While and do while loops - indefinite loops

For loop is definite…

For (initialization exp; condition exp; incre or decre exp)

Ans:

import java.util.Scanner;

public class LoginValidation {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Using while loop

String loginid = "user123";

String pwd = "pass123";

int count = 0;

while (!loginid.equals("user123") || !pwd.equals("pass123")) {

System.out.println("You have logged in for " + (++count) + " times");

System.out.println("Enter your login id:");

loginid = sc.nextLine();

System.out.println("Enter your password:");

pwd = sc.nextLine();

}

// Using do-while loop

count = 0;

do {

System.out.println("You have logged in for " + (++count) + " times");

System.out.println("Enter your login id:");

loginid = sc.nextLine();

System.out.println("Enter your password:");

pwd = sc.nextLine();

} while (!loginid.equals("user123") || !pwd.equals("pass123"));

sc.close();

}

}

Task 13:

Wap to display numbers from 10 to 1 .. skip 7 and 5.

for(int i= 10; i >0; i–){

If ( i == 5 || i == 7)

Continue;

sout(i);

}

Ans:

public class SkipNumbers {

public static void main(String[] args) {

for(int i = 10; i > 0; i--) {

if(i == 5 || i == 7)

continue;

System.out.println(i);

}

}

}

Task 014:

Arrays:

Try the below code and display the output…

Now play with it try to access arr of 5th index and see the output…and try to access arr of -1 index and see the output..

package Arrays;

public class Demo01 {

public static void main(String[] args) {

// TODO Auto-generated method stub

char[] arr = {'a','e','i','o','u'};

System.out.println(arr);

String[] names = {"Meena", "Tina", "Veena", "heena"};

System.out.println(names[0]);

names[1]= "Reena";

System.out.println(names[1]);

System.out.println(names.length);

System.out.println(names[4]);

//Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException

}

}

Ans:

public class ArrayDemo {

public static void main(String[] args) {

char[] arr = {'a','e','i','o','u'};

System.out.println(arr);

String[] names = {"Meena", "Tina", "Veena", "heena"};

System.out.println(names[0]);

names[1] = "Reena";

System.out.println(names[1]);

System.out.println(names.length);

// This will cause ArrayIndexOutOfBoundsException

// System.out.println(names[4]);

}

}

Strings:

Task 015:

String – non primitive data type —> collection of characters or any value within “ ”

– immutable ⇒ cannot be changed

String Name = “Prasunamba is a trainer”;

Name = “Hello”;

Variables are mutable ⇒ which can be changed

package StringHandling;

public class Demo01 {

public static void main(String[] args) {

// TODO Auto-generated method stub

String str1 = "Java Strings "; // string Literal

String str2 = new String(str1); // obj of the string - new keyword

String str3 = new String("are easy to learn ");

char ch[] = {'S', 't', 'r' ,'i', 'n', 'g'};

String str4 = new String(ch);

System.out.println(str1 + "\n" + str2 + "\n" +str3 + "\n" +str4);

}

}

public class StringDemo {

public static void main(String[] args) {

String str1 = "Java Strings "; // string Literal

String str2 = new String(str1); // object of the string

String str3 = new String("are easy to learn ");

char ch[] = {'S', 't', 'r', 'i', 'n', 'g'};

String str4 = new String(ch);

System.out.println(str1 + "\n" + str2 + "\n" + str3 + "\n" + str4);

}

}

Task 016

Enums or Enumerations – part of collection framework

What is the output of the below code snippet

package Enumerations;

enum color{

red, blue, green, yellow

}

public class Demo01 {

public static void main(String[] args) {

color c1 = color.yellow;

System.out.println(c1);

}

}

package Enumerations;

enum Weekdays{

Sunday , Monday , Tuesday

}

public class Demo01 {

public static void main(String[] args) {

Weekdays c1 = Weekdays.Tuesday;

System.out.println(c1);

}

}

enum Color {

RED, BLUE, GREEN, YELLOW

}

public class EnumDemo {

public static void main(String[] args) {

Color c1 = Color.YELLOW;

System.out.println(c1);

}

}

Task 017:

Getter and setter

Create a program name Person.java

public class Person {

private String name;

// Getter

public String getName() {

return name;

}

// Setter

public void setName(String newName) {

this.name = newName;

}

}

Create another program named Task017.java

public class Task017{

public static void main(String[] args) {

Person myObj = new Person();

myObj.name = "John";

System.out.println(myObj.name);

}

}

—----------------------------------what is the reason for the error —---------------explain

Ans:

The error in Task017 occurs because we're trying to access the private field 'name' directly. In Java, private members can only be accessed within the same class. That's why we use getters and setters to access private fields from outside the class.

// Person.java

public class Person {

private String name;

// Getter

public String getName() {

return name;

}

// Setter

public void setName(String newName) {

this.name = newName;

}

}

Task 018

Now create one more program named Task018.java

public class Main {

public static void main(String[] args) {

Person myObj = new Person();

myObj.setName("John");

System.out.println(myObj.getName());

}

}

Now —--------------think what is the output of the above code—--------------

public class Task018 {

public static void main(String[] args) {

Person myObj = new Person();

myObj.setName("John");

System.out.println(myObj.getName());

}

}

=========================================================================================================================================================================

public class Task015{

public static void main(String[] args){

String Str1 = "Myname ";

Str1 = "java";

String str2 = "hello";

System.out.println(Str1);

System.out.println(str2);

}

}

//string - immutable

//strings with the same content share storage ina single pool to minimize creating the copy of the same value again and again..

//a string is created/ generated , its content cannot be changed

// string are considered as classes -===> we create objects to it..

//cannot be changed one initialized in string

// variable =---> mutable

// age = 10;

// age = 11;

class Customer01{

void accept(){

System.out.println("Accept method");

}

void display(){

System.out.println("display method");

}

}

public class Task111{

void method1(){

System.out.println("method1 called");

}

static void static\_method1(){

// method1(); ==> cannot access as it is a non static member

System.out.println(" static\_method1 called");

}

public static void main(String[] args){

Customer01 cobj = new Customer01();

cobj.accept();

cobj.display();

static\_method1();

Task111 tobj = new Task111();

tobj.method1();

// method1();

}

}

Task016\_1.java

Enums – understand the code

//Attaching Multiple values

public enum Element {

H("Hydrogen", 1, 1.008f),

HE("Helium", 2, 4.0026f),

// ...

NE("Neon", 10, 20.180f);

private static final Map<String, Element> BY\_LABEL = new HashMap<>();

private static final Map<Integer, Element> BY\_ATOMIC\_NUMBER = new HashMap<>();

private static final Map<Float, Element> BY\_ATOMIC\_WEIGHT = new HashMap<>();

static {

for (Element e : values()) { //for each loop

BY\_LABEL.put(e.label, e);

BY\_ATOMIC\_NUMBER.put(e.atomicNumber, e);

BY\_ATOMIC\_WEIGHT.put(e.atomicWeight, e);

}

}

public final String label;

public final int atomicNumber;

public final float atomicWeight;

private Element(String label, int atomicNumber, float atomicWeight) {

this.label = label;

this.atomicNumber = atomicNumber;

this.atomicWeight = atomicWeight;

}

public static Element valueOfLabel(String label) {

return BY\_LABEL.get(label);

}

public static Element valueOfAtomicNumber(int number) {

return BY\_ATOMIC\_NUMBER.get(number);

}

public static Element valueOfAtomicWeight(float weight) {

return BY\_ATOMIC\_WEIGHT.get(weight);

}

}

Task019.java

Wap to display the content of the above enum from the class Task016\_1 in this program.. (main needs to be added)

Ans:

public class Task019 {

public static void main(String[] args) {

// Display element details

System.out.println("Element Details:");

// Getting element by label

Element hydrogen = Element.valueOfLabel("Hydrogen");

System.out.println("Found by label: " + hydrogen);

// Getting element by atomic number

Element helium = Element.valueOfAtomicNumber(2);

System.out.println("Found by atomic number: " + helium);

// Display all elements

System.out.println("\nAll Elements:");

for (Element e : Element.values()) {

System.out.printf("Element: %s, Atomic Number: %d, Atomic Weight: %.4f%n",

e.label, e.atomicNumber, e.atomicWeight);

}

}

}

Arrays

Task 020:

Create an array of your name

Hint : use

Char[] Name = {‘P’, “r’, ….}; // initializing an array

sout(Name);

Int n = Name.length; // size of your name

sout(“there are “+ n +”letters in my name”);

Use for loop to display each letter..

HInt: use ghe below code snippet…

// traversing array

for (int i = 0; i < n; i++)

System.out.print(Name[i] + " ");

Ans 20:

public class NameArray {

public static void main(String[] args) {

char[] name = {'J', 'O', 'H', 'N'};

// Direct print

System.out.println(name);

// Length of name

int n = name.length;

System.out.println("There are " + n + " letters in my name");

// Display each letter using for loop

System.out.println("Letters in my name:");

for (int i = 0; i < n; i++) {

System.out.print(name[i] + " ");

}

}

}

sHALLOW copy and Deep copy

| **Copy Type** | **Description** | **Example** |
| --- | --- | --- |
| **Shallow Copy** | Both arrays share references to the same objects | **Shallow Copied Array :** [[obj1, obj2] , [obj3], [obj4]] |
| **Deep Copy** | New instances of objects are created. | **Deep copied array :** [[newObj1, newObj2] , [newObj3], [newObj4]] |

public class ArrayCopyDemo {

public static void main(String[] args) {

// Shallow Copy Example

System.out.println("Shallow Copy Example:");

int[][] original = {{1, 2}, {3, 4}};

int[][] shallowCopy = original.clone();

// Modifying shallow copy affects original

shallowCopy[0][0] = 5;

System.out.println("Original array after shallow copy modification:");

printArray(original);

// Deep Copy Example

System.out.println("\nDeep Copy Example:");

int[][] original2 = {{1, 2}, {3, 4}};

int[][] deepCopy = new int[original2.length][];

// Creating deep copy

for(int i = 0; i < original2.length; i++) {

deepCopy[i] = original2[i].clone();

}

// Modifying deep copy doesn't affect original

deepCopy[0][0] = 5;

System.out.println("Original array after deep copy modification:");

printArray(original2);

}

private static void printArray(int[][] arr) {

for(int[] row : arr) {

for(int num : row) {

System.out.print(num + " ");

}

System.out.println();

}

}

}

Key Points to Remember:

Shallow Copy:

* Creates a new array but shares the same references
* Changes in copied array affect original array
* Faster but less secure

Deep Copy:

* Creates new array with new references
* Changes in copied array don't affect original
* Slower but more secure
* When to use:

Shallow Copy: When memory is a concern and you need a quick reference

Deep Copy: When you need completely independent copies of data