**Day-01**

**28-02-2025**

**================**

**Q-1: Do we able to develop the java program without main() method.**

**======================================**

Yes, we can develop the java program without main().

Here, java compiler cannot stop the program from compilation.

JVM (Java Virtual Machine) --> As interpreter

on the byte code, JVM can start to search about "main()" method.

If the main() method is identified, then that program can proceed for the execution. Otherwise, the program can be stopped from the execution and returns a run-time error.



class TestProgram{

 public static void ravi(String[] args)

 {

 System.out.println("Hello");

 }

}

**Q-2:What is the Java program Execution process?**

**===============================**

javac --> java compilation --> compiler

output of the javac ---> java -----> interpreter

source program (.java) --------> compiler -------> Byte code (.class file) -----> interpreter ----> object code (.obj) ----> Platform ---> output

**Q-3: How the java program is platform independent?**

**==================================**

-> After the compilation, the compiler can provide "byte code" that code can be processed towards the interpreter by using "java" command. Then the interpreter interprets the byte code and generate the object code based on the platform on which we can interpret.

**Q-4: How the java program is processor independent?**

**===================================**

-> When we can do the compilation (using javac command), the compiler can generate "byte code" which is completely processor dependent. Which can be generated based on the processor on which we can compile.

**Control Statements:**

**==============**

**Sequential Execution:**

**-------------------------**

-> This can be considered to make execute the java program line by line or statement by statement.

-> The default action while the java program execution is "sequential execution".

**Why control statement?**

**----------------------------**

1) when we want to execute only the selected block of code from multiple blocks of code of the particular application,

2) when we want to execute the particular block of code repeatedly for more than one time and

3) when we want to jump from one block of code to another block code within the application

 we can use "control statements".

**What are types of control statements?**

**---------------------------------------------**

-> There are the three types of control statements:

 1) Conditional statements/Selection statements

 2) Loop Statements/Iterative Statements

 3) Jump Statements/Transfer Statements

**1) Conditional statements/Selection statements**

**---------------------------------------------------------**

-> There are the five different types of conditional statements:

 1) Simple if statement

 2) if else statement

 3) if else if else ladder statement

 4) Nested if else statement

 5) switch statement

**Simple if statement:**

**------------------------**

Syntax:

 if(condition/expression)

 {

 // if-block

 statement-1;

 statement-2;

 }

 next statement;



**// WAP TO CONVERT THE GIVEN NUMBER INTO POSITIVE IF IT IS ENETERED AS NEGATIVE.**

import java.util.Scanner;

class NegativeToPositive{

 public static void main(String[] args)

 {

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter the value:");

 int x = scan.nextInt();

 if(x < 0){

 x = -x;

 }

 System.out.println("The number after the conversion = "+x);

 }

}

**Day-02**

**01-03-2025**

**==================**

if-else statement:

--------------------

Syntax:

 if(condition){

 if-block

 statement-1;

 statement-2;

 }

 else{

 else-block

 statement-3;

 statement-4;

 }

 next statement;

-> When we want to select one particular block of statements out of two block of code, we can use "if-else".

-> The flow of control can check the condition which is defined with if:

 if the condition is "true", then: the control can execute the "if-block" statements after that the control go to the "next statement" to execute.

 otherwise, then: the control execute the "else-block" statements and then go to "next statement" to execute.



**// WAP TO CHECK WHETHER THE GIVEN NUMBER IS POSITIVE OR NEGATIVE.**

/\*

number > 0 ==> positive number

number < 0 ==> negative number

in this task:

 input -> number

 output -> message

\*/

import java.util.Scanner;

class NumberCheck{

 public static void main(String[] args){

 // creating the object for the Scanner class.

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter a number:");

 int number = scan.nextInt();

 // checking the number

 if(number > 0){

 System.out.println(number+" is positive");

 }

 else{

 System.out.println(number+" is negative");

 }

 }

}

**// WAP TO CONVERT THE NUMBER TO POSITIVE IF IT IS ENETERED AS NEGATIVE AND VICEVERSA.**

/\*

input : number

output : message

number > 0 ==> positive

 we should convert into negative

number < 0 ==> negative

 we should convert into positive

\*/

import java.util.Scanner;

class NumberConversion{

 public static void main(String[] args)

 {

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter a number:");

 int number = scan.nextInt();

 if(number > 0){

 number = -number;

 System.out.println("The number " + number+" is converted into negative");

 }

 else{

 number = -number;

 System.out.println("The number " + number+" is converted into positive");

 }

 }

}

**// WAP TO CHECK WHETHER THE GIVEN NUMBER IS EVEN NUMBER OR ODD NUMBER.**

/\*

input: number

output: message

Even number: When the number evenly divided with '2' is called as "Even number".

Evenly dividing ==> when the remainder during the division is '0'

remainder ==> %

Odd number: when the number is not evenly divided with '2'

Syntax for remainder calculation:

 remainder = number % 2

 remainder == 0 ==> even

 remainder != 0 ==> odd

\*/

import java.util.Scanner;

class EvenOrOddCheck{

 public static void main(String[] args)

 {

 Scanner s = new Scanner(System.in);

 System.out.println("Enter the number:");

 int number = s.nextInt();

 int remainder = number % 2;

 if(remainder == 0){

 System.out.println(number + " is an even number");

 }

 else{

 System.out.println(number + " is an odd number");

 }

 }

}

**// WAP TO FIND THE BIGGEST NUMBER AMONG TWO INTEGERS.**

/\*

99, 77

 99 > 77

 99 => big

101, 99 ==> 101 big

 101 > 99

\*/

import java.util.Scanner;

class FindingBig{

 public static void main(String[] args)

 {

 Scanner s = new Scanner(System.in);

 System.out.println("Enter the first value:");

 int a = s.nextInt();

 System.out.println("Enter the second value:");

 int b = s.nextInt();

 int big = 0;

 if(a > b){

 big = a;

 }

 else{

 big = b;

 }

 System.out.println("The Biggest number " + big);

 }

}

Assignment:

===========

1) WAP TO FIND THE SMALLEST NUMBER AONG TWO INTEGERS.

2) WAP TO FIND THE SUM OF TWO NUMBERS WHEN THE FIRST NUMBER IS LESS THAN TO THE SECOND NUMBER. WHEN THE FIRST NUMBER IS GREATER THAN THE SECOND NUMBER, THEN YOU CAN FIND THE SUBTRACTION OF THE GIVEN TWO NUMBERS.

**Day-03**

**03-03-2025**

**===============**

**/\* WAP TO FIND THE SMALLEST VALUE AMONG TWO GIVEN INTEGERS**

**Logic:**

 **input: two integers**

 **output: message**

**n1, n2**

 **n1 < n2:**

 **n1 is small**

 **n2 < n1:**

 **n2 is small**

**\*/**

import java.util.Scanner;

class FindingSmallerValue{

 public static void main(String[] args)

 {

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter the first value:");

 int n1 = scan.nextInt();

 System.out.println("Enter the second value:");

 int n2 = scan.nextInt();

 int smaller = 0;

 if(n1 < n2){

 smaller = n1;

 System.out.println("The Smallest number is = "+smaller);

 }

 else{

 smaller = n2;

 System.out.println("The Smallest number is = "+smaller);

 }

 }

}

**/\* WAP TO FIND THE SUM OF TWO INTEGERS WHEN THE FIRST NUMBER IS LESS THAN THE SECOND NUMBER. OTHERWISE FIND THE DIFFERENCE.**

**INPUT: N1, N2 ARE TWO INTEGERS**

**OUTPUT: SUM OR DIFFERENCE**

**N1 < N2 ==> SUM**

**N1 > N2 ==> DIFFERENCE**

**\*/**

import java.util.Scanner;

class FindingSumOrDifference{

 public static void main(String[] args)

 {

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter the first value:");

 int n1 = scan.nextInt();

 System.out.println("Enter the second value:");

 int n2 = scan.nextInt();

 if(n1 < n2){

 int sum = n1 + n2;

 System.out.println("The Sum of given two numbers = "+sum);

 }

 else{

 int difference = n1 - n2;

 System.out.println("The Difference of given two numbers = "+difference);

 }

 }

}

**/\* WRITE A PROGRAM USING JAVA**

**ASSUME USERNAME AS "admin"**

**AND PASSWORD AS "admin123".**

**Now, check whether the user has successfully logged in or not.**

**\*/**

import java.util.Scanner;

class LoginCheck{

 public static void main(String[] args)

 {

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter the user name:");

 String username = scan.nextLine();

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

 String password = scan.nextLine();

 if(username.equals("admin") && password.equals("admin123")){

 System.out.println("Login Success");

 }

 else{

 System.out.println("Login Fail");

 }

 }

}

Note:

-----

 1) when we can apply '==' on two strings for the comparison:

 '==' operator can compare two addresses of strings.

 2) to compare the two strings with internal content (data), we can use "equals()".

 Syntax:

 str1.equals(str2)

class Dummy{

 public static void main(String[] args)

 {

 String s1 = "Java";

 String s2 = "java";

 String s3 = s1;

 System.out.println(s1 == s3);

 System.out.println(s1 == s2);

 System.out.println(s1.equals(s2));

 System.out.println(s1.equalsIgnoreCase(s2));

 System.out.println(123 == 123);

 }

}

 3) '==' compares the values of integer values only.

**if else if else ladder:**

**=============**

Syntax:

 if(condition1){

 block-1;

 }

 else if(condition2){

 block-2;

 }

 else if(condition3){

 block-3;

 }

 else{

 block-n;

 }



**// WAP TO CHECK WHETHER THE GIVEN NUMBER IS POSITIVE OR NEGATIVE OR ZERO.**

import java.util.Scanner;

class NumberCheck{

 public static void main(String[] args){

 Scanner s = new Scanner(System.in);

 System.out.println("Enter a value:");

 int n = s.nextInt();

 if(n == 0){

 System.out.println("The number is zero");

 }

 else if(n > 0){

 System.out.println("The number is positive");

 }

 else{

 System.out.println("The number is negative");

 }

 }

}

**// WAP TO FIND THE BIGGEST NUMBER AMONG FIVE INTEGERS.**

import java.util.Scanner;

class FindingBiggest{

 public static void main(String[] args)

 {

 Scanner s = new Scanner(System.in);

 System.out.println("Enter 5 integers:");

 int a = s.nextInt(); // 121

 int b = s.nextInt(); // 131

 int c = s.nextInt(); // 142

 int d = s.nextInt(); // 200

 int e = s.nextInt(); // 299

 if(a > b && a > c && a > d && a > e){

 System.out.println("a is biggest number among the given 5 numbers");

 }

 else if(b > c && b > d && b > e){

 System.out.println("b is biggest number among the given 5 numbers");

 }

 else if(c > d && c > e){

 System.out.println("c is biggest number among the given 5 numbers");

 }

 else if(d > e){

 System.out.println("d is biggest number among the given 5 numbers");

 }

 else{

 System.out.println("e is biggest number among the given 5 numbers");

 }

 }

}

**Assignment:**

**=========**

1) WAP TO FIND THE SMALL NUMBER AMONG FOUR INTEGERS.

2) WAP TO FIND WHETHER THE PERSON IS ELIGIBLE TO VOTE OR NOT.

3) WAP TO ACCEPT 5 SUBJECT MARKS AS INPUT. AND CALCULATE THE STUDENT GRADE BY FOLLOWING THE GIVEN CONDITIONS:

 percentage >= 85 ==> "Grade-A"

 percentage >= 70 and percentage < 85 ==> "Grade-B"

 percentage >= 55 and percentage < 70 ==> "Grade-C"

 percentage >= 40 and percentage < 55 ==> "Grade-D"

 percentage < 40 ==> "Fail"

**Day-04**

**04-03-2025**

**===============**

**// Student Grade System**

class StudentGradeSystem{

 public static void main(String[] args)

 {

 int s1, s2, s3, s4, s5; // variable declaration

 s1 = 77;

 s2 = 88;

 s3 = 67;

 s4 = 91;

 s5 = 89;

 int total = s1 + s2 + s3 + s4 + s5;

 int percentage = total/5;

 System.out.println("The Total marks = "+total);

 System.out.println("Percentage = "+percentage);

 char grade = 0;

 if(percentage >= 85){

 grade = 'A';

 }

 else if(percentage >= 70 && percentage < 85){

 grade = 'B';

 }

 else if(percentage >= 55 && percentage < 70){

 grade = 'C';

 }

 else if(percentage >= 40 && percentage < 55){

 grade = 'D';

 }

 else{

 System.out.println("Fail");

 }

 System.out.println("Grade of the student = "+grade);

 }

}

**Block variable:**

**==========**

-> The variable which can declare and assign with value within the block and can be allowed to access within the same block only. If we can try to access in outside of the defined block, we can get "compile-time error".

Ex:

if(condition){

 int a;

 a = 12;

 Sop(a);

}

else{

 Sop(a); ==> error

}

Note:

-----

When we need to perform the re-assignment of the variable again and again after the assignment, we need to initialize the variable with some default value according to the specified type.

Ex: int a = 0;

**Nested if else:**

**==========**

-> when a block defining within the another block, we can call it as "nested block".

Ex:

block1{

 statements;

 block2{

 statements;

 }

}

-> nested if else describing that writing/defining if else in another if else blocks.

Syntax:

 if(condition1){

 statements;

 if(condition2){

 statements;

 }

 else{

 statements;

 }

 }

 else{

 if(condition3){

 statements;

 }

 else{

 statements;

 }

 }

 next statement;

**// WAP TO FIND THE SMALLEST NUMBER AMONG THREE INTEGERS.**

import java.util.Scanner;

class SmallestNumberFromThree{

 public static void main(String[] args){

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter three integers:");

 int a = scan.nextInt(); // 29

 int b = scan.nextInt(); // 19

 int c = scan.nextInt(); // -39

 if(a < b){

 if(a < c){

 System.out.println(a + " is smallest number");

 }

 else{

 System.out.println(c + " is smallest number");

 }

 }

 else{

 if(b < c){

 System.out.println(b + " is smallest number");

 }

 else{

 System.out.println(c + " is smallest number");

 }

 }

 }

}

**Day-05**

**05-05-2025**

**================**

if(condition1){

 if(condition2){

 block1;

 statements;

 }

 else{

 block2;

 statements;

 }

}

else

{

 if(condition3)

 {

 block3;

 statements;

 }

 else{

 block4;

 statements;

 }

}

next statement;

**/\* WAP USING JAVA TO CHECK WHETHER THE GIVEN YEAR IS LEAP YEAR OR NOT**

**year = 20;**

**1) if year is successfully divided with '4' (by giving the remainder as '0')**

**==> that year is a leap year.**

**% ==> remainder after the division**

**20 % 4 ==> 0 ==> Leap year**

**35 % 4 != - ==> not a leap year**

**2) the leap year is not to be successfully divided with '100'**

**But some years which are divisible with '100' also consider as leap year.**

**Ex: 400, 800, 1200, 1600, 2000 etc.**

**when the year is divisible with 100 and also divisible with '400' ==> leap year**

**otherwise ==> not a leap year**

**\*/**

import java.util.Scanner;

class LeapYearCheck{

 public static void main(String[] args)

 {

 Scanner scan = new Scanner(System.in);

 System.out.println("Enter the year:");

 int year = scan.nextInt();

 if(year % 4 == 0){

 if(year % 100 != 0){

 System.out.println(year+" is leap year");

 }

 else{

 if(year % 400 == 0){

 System.out.println(year + " is leap year");

 }

 else{

 System.out.println(year + " is not a leap year");

 }

 }

 }

 else{

 System.out.println(year + " is not a leap year");

 }

 }

}

**/\* WAP TO TAKE AN INTEGER AS AN INPUT AND DO THE FOLLOWING:**

**i) IF IT IS DIVISIBLE BY '3' THEN DISPLAY 'HI'**

**ii) IF IT IS DIVISIBLE BY '5' THEN DISPLAY "HELLO"**

**iii) IF IT IS DIVISIBLE BY 3 AND 5 BOTH, THEN DISPLAY "BYE BYE".**

**\*/**

import java.util.Scanner;

class PracticeProblem1{

 public static void main(String[] args)

 {

 Scanner s = new Scanner(System.in);

 System.out.println("Enter a value:");

 int n = s.nextInt(); // 6 10 15 9

 if(n % 3 == 0 && n % 5 == 0){

 System.out.println("Bye");

 }

 else if(n % 3 == 0){

 System.out.println("Hi");

 }

 else if(n % 5 == 0){

 System.out.println("Hello");

 }

 else{

 System.out.println("Number is not divided with 3 and 5 both");

 }

 }

}

**/\* WAP TO MULTIPLY THREE INTEGERS HOWEVER, IF ONE OF THE NUMBER IS '7' THEN CONSIDER THE VALUES RIGHT TO IT FOR MULTIPLICATION. IF '7' IS THE LAST NUMBER THEN DISPLAY THE OUTPUT AS '-1'.**

**a, b and c**

 **==> a \* b \* c**

**if a == 7**

 **multiplication ==> b \* c**

**if b == 7**

 **multiplication ==> c**

**if c == 7**

 **multiplication ==> -1**

**\*/**

import java.util.Scanner;

class PracticeProblem2{

 public static void main(String[] args)

 {

 Scanner s = new Scanner(System.in);

 System.out.println("Enter three integers:");

 int a = s.nextInt();

 int b = s.nextInt();

 int c = s.nextInt();

 int result;

 if(a == 7){

 result = b \* c;

 }

 else if(b == 7){

 result = c;

 }

 else if(c == 7){

 result = -1;

 }

 else{

 result = a \* b \* c;

 }

 System.out.println("The multiplication Result = "+result);

 }

}