

Chapter - 2.

Elementary Programming

- (•) Primitive data types.
- (•) Variables.
- (•) Constants.
- (•) operators.
- (•) expression.
- (•) Input and Output.

Writing a Simple program

- ① First Right the algorithm then convert into programming language.
- ② Algorithm :- (Pseudo Code)
- (•) Every Java program Start with class name.

```
Public Class Circle {  
    Public Static Void main(String[] args) {  
        3  
        4  
    }  
}
```

① Storing the radius in the program

For storing any element you need to declare a variable.

A Variable represents a Value Stored in the Computer memory.

Don't use x and y as Variable names.

Use descriptive names
radius for radius.
area for area.

To know Compiler what Variable is
they use data types

(char)

(int) ← integer ; Character; boolean ⇒ primitive.

Floating point (double).

To display the result.

```
System.out.println("The area of the circle  
+ radius + " + area);
```

① Reading Input from the Console.

```
import java.util.Scanner;
```

```
Scanner input = new Scanner(System.in);
```

```
double radius = input.nextDouble();
```

• Identifiers:

⇒ Identifiers are the names that identify the element such as classes, methods and variables in a program.

eg:- ComputeAverage, main, input, etc.
the names of things that appear in the program.

In programming terminology Such names are called Identifiers.

• Variables:

⇒ To Store a Value in a program we use Variable.

Variable is like a container that stores Value.

In Variable Values can be changed. So it called Variable.

datatype VariableName j;

(byte, short, long, float, char, boolean etc).

① Assignment Statements and Assignment Expression.

To Give Value to a Variable. Use like.

Assignment Statements.

Variable = expression;

② Named Constants.

A Named Constant is an Identifier that represents a permanent value.

Declared Const name

final datatype (const name = value);

e.g:-

final double PI = 3.14;

Numeric Data types and Operations.

⇒ Every data types has a range of values

1. byte -2^7 to 2^7 (8 bit) byte type
 2. Short -2^{15} to 2^{15} (16 bit).
 3. int -2^{31} to 2^{31} (32 bit)
 4. long -2^{63} to $2^{63}-1$ (64 bit)
(32 bit IEEE 754)
 5. float. (-ve). - $3.402E+38$ to $-1.4E-45$.
(+ve) = $1.4E$ to $3.402E+38$.
 6. Double (+ve) (64 bit IEEE 754)
(-ve)

Reading Numbers from the keyboard.

next Byte(), eg. byte byteValue = input.next Byte();

next Short().

next Int().

next Double().

Numeric Operators.

(+) Addition.

(-) Subtraction.

(*) Multiplication.

(/) Division.

(%.) Remainder.

① Exponent operations.

Math·pow(a,b). = a^b .

e.g :-

System.out.println(math·pow(2,3)).

- ② literals. \Rightarrow A literal is a constant value that appears directly in a program.

Operators Precedence.

① + , - , * , / .

② + - :

if. left \rightarrow Right.

Augmented Assignment Operators

$\text{Count} = \text{Count} + 1$

$\text{Count} += 1$

$+ = \quad i + 8 \quad , \quad i = i + 8$

$- = \quad i - 8 \quad , \quad i = i - 8$

$* = \quad i * 8 \quad , \quad i = i * 8$

Increment Decrement Operators

$i++ \quad i + i = 3$

$++i \quad$

Chp-3. Selections.

- ① The program can decide which statements to execute based on a condition.
- ② boolean Data Types.

< (less than).

<= (less than or equal to)

> greater than.

>= greater than or equal to

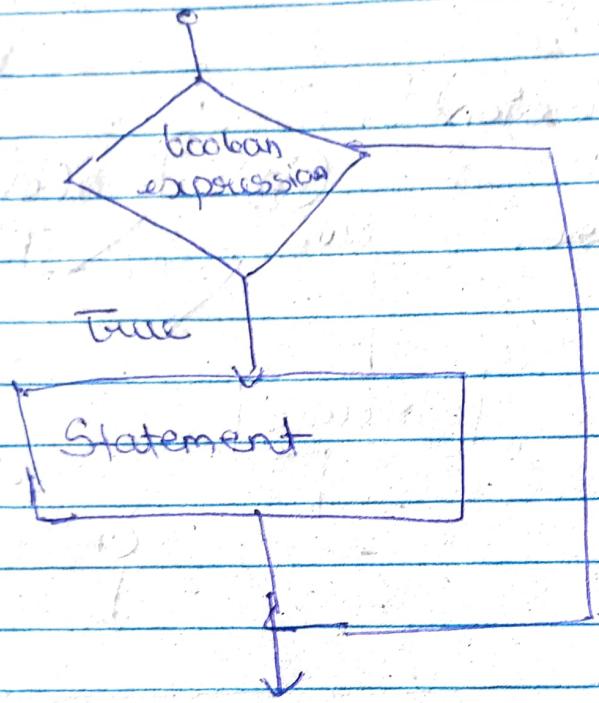
= = equal to.

!= not equal to

④ Selection.

if, if else, nested if, multiple if-else,
switch statements

if - Statement



must Syntax

if() { }
not having
braces is
not necessary

Two way if - else statements.

if (boolean-expression) {

true case.

{

else {

{

false case.

(if)

(else)

True

...

Con

False

true case

false case

Nested if

if ().

else if ().

else if ().

else

Generating Random Numbers.

Math.random()

e.g -

(int)(Math.random() * 10);

(int)(Math.random() * 10);

↳ Values from
0 to 9.

Logical operators.

!, &, ||, and ^

||

and

or

T	F
F	T

$$\begin{cases} T \& T = T \\ T \& F = F \\ F \& T = F \\ F \& F = F \end{cases}$$

$$\begin{cases} T \& T = T \\ T \& F = F \\ F \& T = F \\ F \& F = F \end{cases}$$

Switch Statements.

A Switch Statement executes statements based on the value of a variable or an expression.

Switch () {

Case 0:

 statements
 break;

Case 1:

 statements
 break;

Case 2:

 statements
 break;

default:

3.

Conditional Expression.

if ($x > 0$) .

$y = 1j$

else

$y = -1j$

$y = (x > 0) ? 1 : -1j$

booban expression ? expression1 : expression2.

L-4. Mathematical Functions, Characters and Strings.

① Common Mathematical function.

e.g:-

Math. Sin(0)

Math. toDegree(Math.PI/2)

② Exponent Methods.

$$\exp(x) = (e^x).$$

$$\log(x) = \text{log}_e(x).$$

$$\log_{10}(x) = \log_{10}(x)$$

$$\text{Pow}(a,b) = (a^b)$$

$$\text{Sqrt}(x) = x(\sqrt{x})$$

③ Rounding Methods.

$\text{ceil}(x) \Rightarrow x$ is rounded up to nearest integer.

$\text{floor}(x) \Rightarrow x$ is rounded down to nearest.

$\text{rint}(x) \Rightarrow x$ is rounded up to its nearest.

$\text{round}(x) \Rightarrow \text{Math.floor}(x+0.5)$.

① Character Data types and operations.

char letter = 'A'

Char numletter = '4'

② Unicode and ASCII Code.

Character

'0' to '9' 48 to 57

'A' to 'Z' 65 to 90

'a' to 'z' 97 to 122

③ Escape Sequences for Special Characters

\b

\t

\n

\f

\r

\\"

\''

① min, max and abs. method.

Math. max(2,3) returns 3.

Math. max(25,3) return 4.

Math. abs(-2) return 2.

② The random Method.

Generate random method.

(int) (Math.random() * 10) → 0 to 9.

50 + (int) (Math.random() * 50) → 50 to 99.

The String Type

Eg:- Reading String from the Console.

```
Scanner input = new Scanner (System.in);  
System.out.print ("Enter your message: ");  
String s1 = input.next();  
String s2 = input.next();  
String s = input.nextLine();
```

Output

Enter Your message: Welcome to Java.

s1 = Welcome.

s2 = Java.

s = Welcome Java.

L-5. LOOPS.

Introduction.

A loop can be used to tell a program to execute statement repeatedly.

① The while loop.

```
int count = 0;
while (count < 100) {
    System.out.println ("Welcome to Java!");
    count++;
}
```

Syntax.

```
while ( ) {
```

Statement(s).

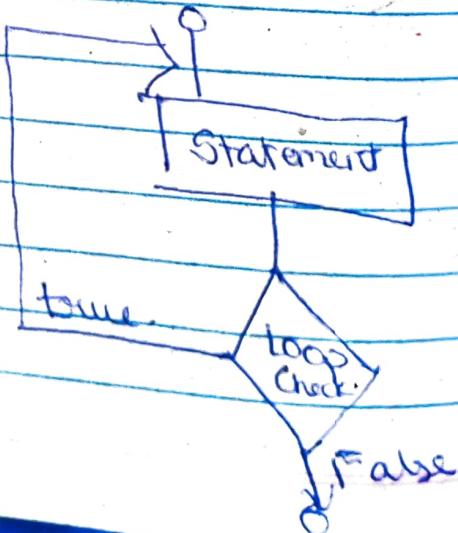
```
}
```

② The do-while loop.

```
do {
```

Statement(s).

```
} while ( );
```



~~①~~. The For Loop:

① $i = \text{Initial Value}$;
while ($i < \text{endValue}$)

$i++$;

3.

① for ($i = \text{Initial Value}; i < \text{endValue}; i++$) { }

3.