**JAVA STUDY MATERIAL**

java is a programming language.

Java is a development environment. (write,test,execute)

java is a deployment environment. (developed code must run in client's environment)

Similar in syntax to c++ similar to semantics to smalltalk.

operating system independent.

Types of Programming Lang:

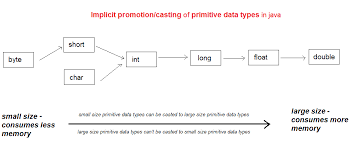
* Structured lang or Procedural Lang - sequential flow of program , step by step lang (C lang,)
* Object Oriented Programming Lang.

High Lvl Lang does not understand user input so it uses translator to translate the user input.

Java is a object oriented programming Lang (OOP).

source code->compiler->object code->Linker->Executable->ML

Java doesnt give object code it gives byte code after compiler.



**Java Variables**

Variables are containers for storing data values.

In Java, there are different **types** of variables, for example:

* String - stores text, such as "Hello". String values are surrounded by double quotes
* int - stores integers (whole numbers), without decimals, such as 123 or -123
* float - stores floating point numbers, with decimals, such as 19.99 or -19.99
* char - stores single characters, such as 'a' or 'B'. Char values are surrounded by single quotes
* boolean - stores values with two states: atrue or false

Declare a Variable:

*type variableName = value;*

int- 4 byte, long- 8 bytes, short- 2 bytes, double – 8 bytes (-128 to 127)

**Literals**:

Literals in Java are a synthetic representation of boolean, character, numeric, or string data. are constant values that directly appear in a program and can be assigned now to a variable.

**C Programming Flow chart:**

A diagram of a computer process

Description automatically generated

**OPERATORS IN JAVA:**

+ - add; - - sub; \* - multiply; / - gives quotient; % - remainder..

**Relational Operators:**  < , > , < = , > = , = = , !=

**Logical Operators:**  AND (&&), OR(||), NOT(!=).

**Ternary Operator:**  for if we use ? for else we use :  **(a>b && a>c ? a : b>a &&b>c ? b : c)**

Features of java:

* Object Oriented
* simple
* Robust (strict checking, error free)
* Architecture neutral
* Portable
* Secure
* High Performance
* Interpreted(Line By Line Conversion & Execution)
* Support for Multi-threading (multiple parallel activities)
* Distributed

**Platform Independence:**

A platform is the hardware & Software Environment in which program runs once compiled java code runs on any platform without recompiling or any kind of modification. "Write once and run everwhere".

JVM directly interacts with Operating System.

**Java Programming Flow chart:**

**A diagram of a program

Description automatically generated**

**Loop Concepts:**

* **While** - used for finite number of events. Decalare a variable at first and then while with condition and print the number and increment.
* **Do While** - It is also used for finiter number of events. Do is written first and block of statements are given , while with condition is given outside the block. ; - is important.
* **For**- It contains of initialization, condition, and incrementation.

**CLASSES AND OBJECTS:**

Class –

Objects – instance of the class

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Day 2 Session**

**MULTIDIMENSIONAL ARRAYS:**

A multidimensional array is an array of arrays. Multidimensional arrays are useful when you want to store data as a tabular form, like a table with rows and columns.

For- each loop is for iterating a variable.

Public: It is an Access modifier, which specifies from where and who can access the method.

Static: It is an Access modifier, which specifies from where and who can access the method.

Void : doesn’t return anything.

Main(): runtime call the main method.

String args[]: command line parameter passed to main method. [] can be allotted after args or string .

**OOPS CONCEPTS:**

* Class & objects- class is a blueprint of the object . Object can be anything.
* Abstraction- hiding the information of the process and explaining which is needed.
* Encapsulation- binding the data in the process which can hide the abstract information.
* Polymorphism-
* Inheritance

**Instantiation relationship:**

A relation Between class and objects.

**Composition relationship:**

A composition in Java between two objects associated with each other exists when there is a strong relationship between one class and another.

**Utilization relationship:**

No memory will be allocated within the class or function.

Java doesn’t support multiple inheritance. Constructor does not have a return type.

Without parameter is called default constructor.