**Core Java**

Java is an object-oriented programming language.

Simple Java program:

**package** com.lokesh;

**Public** **class** HelloWorld {

**public** **static** **void** main(String[] rags) {

// **TODO** Auto-generated method stub

System.***out***.println("Hello World");

}

}

Class – keyword to declare class

Public – access modifier…visible to all

Static – keyword….there is no need to create an object to invoke the static object

Void – return type of method……… doesn’t return any value

Main – starting point of program

String[] args – used for command line arguments

System.out.println() – used to print the statement

Platforms in Java

JAVA SE (Java Standard Edition) – includes API’s like java.lang, java.io, java.util, java.sql, java.math. includes core concepts like OOPS, String, Regex, Exception, Inner classes, Multithreading, Swing, Collection etc

JAVA EE (Java Enterprise Edition) - used for Servlet, JSP, Web Services , JPA etc

JAVA ME (Java Micro Edition) – mobile applications

JAVA FX – rich internet applications. Uses lightweight user interface API.

Java code (Simple.Java)------- Compiler --------------- Byte code(Simple.class)

Java Variables:

Container which holds the value.

A variable is assigned with data type

Three types of variables –

Local – variable declared inside the body

instance – declared inside the class outside the body, not declared as static

Static – variable declared as static is called static variable

Example:

**public** **class** HelloWorld {

**static** **int** *a*=10; //static variable

**public** **void** method() {

**int** b =20; //local variable

System.***out***.println(b);

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** c =30; //instance variable

System.***out***.println(*a*);

HelloWorld hw =**new** HelloWorld();

hw.method();

System.***out***.println(c);

}

}

Simple program of addition of two numbers:

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** a =10;

**int** b =20;

**int** c = a + b;

System.***out***.println(c);

}

}

Typecasting (also called as widening): it’s like a bigger data type is changed to smaller data type

**Package** com.lokesh;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**float** a =10.5f;

**int** b =(**int**) a;

System.***out***.println(a);

}

}

The other is widening like smaller data type is changed to bigger data type then casting is not required you can simply initialise to different data type directly

Data Types:

Different sizes (int, byte, float…) and values that can be stored in variable

Two types - Primitive and non-primitive

Primitive (Boolean (1bit), char (2 byte), byte (1 byte) ,short (2 byte), int (4 byte), long (8 byte), float (4 byte), double (8 byte))

Example:

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Boolean a =**true**;

**byte** b =10;

**char** c ='a';

**short** d =5;

**int** e =10;

**long** f =100;

**float** g = 2.34f;

**double** h = 20.2;

}

}

Non-primitive – String, Array …etc

Example:

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

String a ="hello";

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

}

}

Scanner class in Java:

To pass values dynamically we use this class. In below for variables a and b we are passing dynamically

Example:

**package** com.lokesh;

**import** java.util.Scanner;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("enter a value");

**int** a = sc.nextInt();

System.***out***.println("enter b value");

String b = sc.next();

System.***out***.println(a);

System.***out***.println(b);

}

}

Operators in Java:

Perform operations like +,-,/,\* etc

Many types of operators : unary (a++. ++a, !) , arithmetic (+,-./,\*), shift(>>,<<), relational(==,!=), bitwise (&,!,), logical (&&,||), ternary (?:), assignment (=)

&& - doesn’t check second condition if first condition is false

& - checks both condition

|| - doesn’t check second condition if first condition is true

Terenary operator (?:) – replacement for if-then-else statement

Example:

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** a =10;

**int** b =20;

**int** c =30;

**boolean** d= **true**;

**int** e = a+b; //arithmetic

**int** f =a++; //Unary Operator

**int** g =++a;

System.***out***.println(!d); // opposite of boolean value

System.***out***.println(a>b && a<c); // false && true = false

System.***out***.println(a>b & a<c); // false && true = false

System.***out***.println(a<b || a>c);

**int** h = (a<b)?b:c; //if condition is true then b is assigned otherwise c assigned

}

}

Control Statements:

Decision making statements – if statements, switch statement

Loop statements – do while, for, while, for-each loop

Jump statements – break, continue

Simple if: if condition true then executes statement

If-else: if condition true then executes if block statement otherwise else block statement executed

If-else-if: if condition true then executes if block statement otherwise elseif condition true then elseif block statement executed and if there more elseif…it checks further

Nested if: multiple if condition or if-else

Example:

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** a =10;

**int** b =20;

**int** c =30;

//if

**if**(a<b) {

System.***out***.println("a is less");

}

//if-else

**if**(a<b) {

System.***out***.println("a is less");

}**else** {

System.***out***.println("a is greater");

}

//if-elseif

**if**(a<b) {

System.***out***.println("a is less");

}**else** **if**(a<c) {

System.***out***.println("a is greater");

}**else** {

System.***out***.println(" a didin't satisfy any above conditions");

}

//nested if

**if**(a<b) {

**if**(a<c) {

System.***out***.println("a is lowest");

}

}

}

}

Switch: contains multiple blocks of code called cases and single case is executed based on variable being switched. Default is optional and when nothing matches this gets executed

**package** com.lokesh;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** a=2;

**switch**(a){

**case** 0:

System.***out***.println("value is 0");

**break**;

**case** 1:

System.***out***.println("value is 1");

**break**;

**case** 2:

System.***out***.println("value is 2");

**break**;

**case** 3:

System.***out***.println("value is 3");

**break**;

**default**:

System.***out***.println("nothing matched");

}

}

}

For loop: initialized the loop variable, checks the condition and increment/decrement

Example:

**package** com.lokesh;

**import** java.util.Scanner;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("enter a value");

**int** a = sc.nextInt();

**for**(**int** b=1;b<=a;b++) {

System.***out***.println(b);

}

}

}

For-each: enhanced for loop it just initialised the value of arrays

Example:

**package** com.lokesh;

**import** java.util.Scanner;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**int** a[] = {0,1,2,3};

**for**(**int** b : a) {

System.***out***.println(b);

}

}

}

While loop : loop condition executes until condition is true

Example:

**package** com.lokesh;

**import** java.util.Scanner;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("enter a value");

**int** a = sc.nextInt();

**int** i = 0;

**while**(i< a) {

System.***out***.println(i);

i++;

}

}

}

Do-While: if wanted execute atleast once then do-while is used , after once executed it checks while condition

Example:

**package** com.lokesh;

**import** java.util.Scanner;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("enter a value");

**int** a = sc.nextInt();

**int** i = 0;

**do** {

System.***out***.println(i);

i++;

}

**while**(i< a);

}

}

Break: used to break current flow of program and transfer to next statement. If nested loop is used it breaks inner loop.

Example:

**package** com.lokesh;

**import** java.util.Scanner;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("enter a value");

**int** a=sc.nextInt();

**int** i =0;

**while**(i<a) {

System.***out***.println(i);

i++;

**if**(i==3) {

**break**;

}

}

}

}

Continue: it doesn’t break the loop, it skips part of loop and jumps to next iteration

Example:

**package** com.lokesh;

**import** java.util.Scanner;

**public** **class** HelloWorld {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("enter a value");

**int** a=sc.nextInt();

**int** b =10;

**int** i =0;

**while**(i<a) {

i++;

**if**(i==3) {

**continue**;

}

System.***out***.println(i); //3 will be skipped to display

}

}

}