Java

The basic syntax of java is

Syntax: public class {

Public static void main (String [] args) {

Execution code

}

}

Eg: package begginer;

public class basic {

public static void main(String[] args) {

// TODO Auto-generated method stub

int a= 2;

int b= 3;

System.out.println(a );

System.out.println(b);

System.out.printf("the values are %d and %d\n", a,b );

System.out.println("the values are "+ a + " and " + b);

}

}

-Different data types present in the java are, these are primitive type

int , short , byte , long , float , double , char , Boolean .

eg : int age = 13;

The non-primitive data type is

String : string here is responsible for not holding the variable but it responsible for referring to that value.

We can use ‘ +’ to concat a string with strings and other data types .

Eg ; String text = ‘hello’

Eg ; sysout(‘helllo my name is ‘ + variable to be concatenated with )

-Loops :

While loop

Syntax : while loop(condition) {

Code

}

For loop

For( initialization ; condition ; incrementing) {

Code

}

Do {

Code

}while (condition)

Conditional statements :

If statement

If (condition) {

}

If and else statement

If (condition) {

} else {

}

If and else if statement

If (condition){

Code

}else if (condition) {

} else{

}

Break statement :if the condition is satisfied, it quits out of the loop .

Switch statement :

Syntax : switch (condition) {

Case ‘value to be compared to ‘:

Action to be taken if it matches.

Break;

Default :

Action to be performed if nothing is matched with the case statement

}

-Getting user input : we should use scanner object to read the value from the keyboard

We should import java.util.scanner;

Syntax : Scanner input(name of the variable ) = new Scanner(system.in)

String line = input.nextLine() (here we are waiting till the user enters the enter key)

The above is the way which we use to read a value from the key .

Here nextLine is used for string and we have more attribute to read different data types from the keys

nextInt , nextDouble to read different values .

-Arrays :

Syntax : int[] values(variable name ) ;

Values = new int[3] ( 3 indicates the length of the array i.e the amount of memory required to store the values in the variable)

Values[0] = 1 ( initializing an array)

Or

Int[] values = {1,2,3} ( this is alternative syntax to declare and initialize the array at the same time)

.length is the array method used to get the length of the array. This is normally used in the loop to iterate over the arrays.

String arrays:

Syntax; String[] values = new String[3];

Values[0] = ‘hello’

Or

String[] values = {‘hello’ , ‘world’ }

To iterate through a string arrays

For ( String new : values ) {

Action to perform but use new variable

}

Multidimensional array:

Int values[][] = new int[2][3] (first value is row and second is columns)

To iterate over 2-dimensioal array

For( i=0 ; i< values.length ; i++) {

For (k =0 ; k<values[i].length; k++) {

Code to be executed

}

}

Class and objects

Class Name {

String name ;

Int age;

Void speak () { (this is a subroutine)

System .out.print(‘hello’)

}

}

In the main class

public Class App {

public static void main (String[] args) {

Name name1 – new Name() (object)

name1.age = 12;

name 1.name= ‘hari’

name1.speak()

-Getters

we can have getter method in the class which we have declared, within that rather than printing the values inside the method which we have declared in the class which we have declared , we can return the value and we can create new variable in the main class and we can use the variable for further execution .

eg: created class

int age () {

totalAge = age+10;

retutn totalAge;

}

Main class

After creating an object

Int values;

Values = object.age()

We can also pass parameter to the methods which are declared for the class which we have created.

-Setter

Setter are used to access the variables which are declared in the new class which we have created. We can make the variables are private so that they are not directly accessible in the main class and we must access it through the method and set the variable this helps not altering the variable.

Created class {

Private age ;

Void setage(int age) {

this.age (here the declared variable and parameter variable are same so we must use this keyword to access the class variable else it will be a scope issue) = age;

}

}

Main class {

Object created

Object.setAge(12) (by this way we are not directly accessing the variable )

-Constructors:

When a new object is created the constructors automatically runs the constructor code

Syntax : created class {

Clalss () {

Code to execute

}

The constructors name should be of the same name as the class name. it can even take the parameter and java will automatically run the proper java constructor.and the constructor does not have a return type .eg of constructor with the parameter is below

class App {

private int a;

public App (int a) {

this.a = a;

System.out.println("hello world "+ a);

}

}

public class Aconstruct {

public static void main(String[] args) {

App app = new App(5);

}

}

Static varialbles : static variables are class variables we can use them inside the main using the created class name

Eg created classA{

Public static name ;

}

Mani class {

classA.name= ‘name to be used’

}

There can one static variable of type name here because there is one class associated with name here . though we may have many objects created for it but we can have only one class variable.