Introduction to Programming and Computer Science

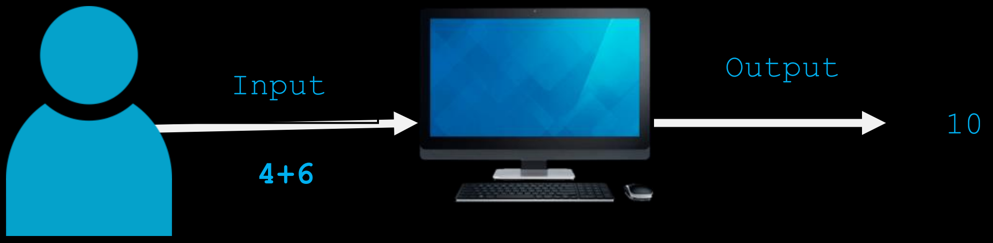
What is Programming?

Programming is a process of telling a computer (Dumb Machine) what task to perform through a set of instructions.

Example: Assume that you have a friend who doesn't know how to cook. One day she came to your house and want to learn cooking from you. So here you will give a set of instructions on how to cook and she will be follow your instructions and perform the tasks. If you gave any instruction wrongly the whole output will be wrong

This is same with the computer. Consider the computer as a non-intelligent friend who doesn't know anything but will perform the task, what you told them to do.

And more importantly it will perform the task more fastly and more accurately than humans, based on the instructions given.



The Language Of Code:

However, telling your computer instructions is not as easy as telling your friend instructions.

While you are interacting with your friend you can use your native language but computer can only understand machine code which consist of binary digits.

Binary digits are 0’s and 1’s.

It is nearly impossible for humans to write a program(set of instructions) only using 0’s and 1’s.

Example: Now consider your friend can only understand hindi and you only know tamil. Here it is impossible for you people to interact with each other without a help of a third person who acts as a translator between two of you. This translator should understand both hindi and tamil.

Compiler:

Compiler is similar to the translator.

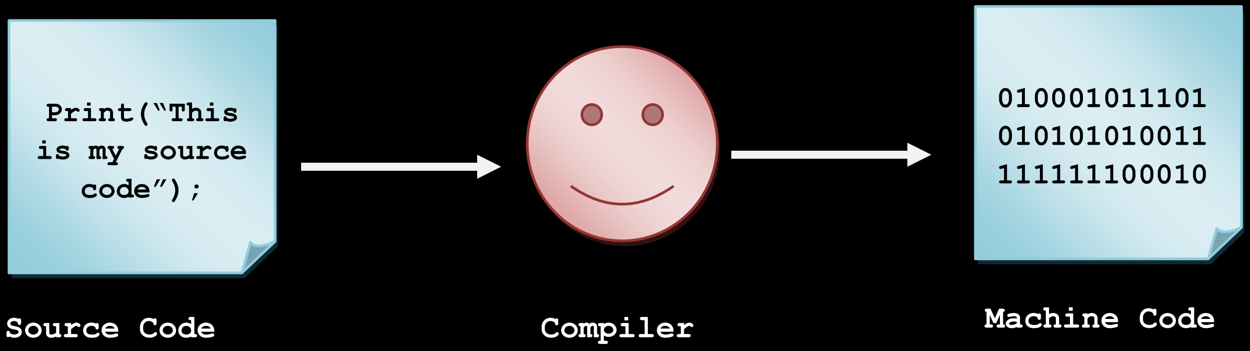
You are writing a set of instructions which is called a program.

It will be written in any high level programming language(Language which will be understood by humans) and given to the compiler(Software program) which will be installed in your computer. It can understand both the human language and the machine code.

This compiler will convert that source code(set of instructions or code that you have written) into machine code which can be easily understood by the computer. After that based on the instructions it will perform the task.

But if there occurs any error during the compilation process. It will throw compile time error and you can’t able to execute the program. This error occurs if there is any wrong syntax present in your program.

The compiler will not allow you to run the program until the errors are removed from the program



```
Print("This
is my source
code");
```

```
010001011101
010101010011
111111100010
```

Source Code

Compiler

Machine Code

Programming Language:

Inorder to get your program compiled by the compiler you should write the set of instructions in any of the programming language. Compiler can only be able to understand the programming language and not your native language.

However learning programming language is not that hard. It can be easily understood by humans.

Programming Languages:

- Each programming language is unique in how they operate.

Java/Python – General purpose programming language

HTML/CSS – Designed for specific task

- Each programming language varies on how powerful it is

JavaScript – scripting language and is used to perform smaller tasks

Java/Python – General purpose languages which is used to perform much more complicated tasks.

- There are also two types of programming languages
  1. High level programming language – Human understandable (Java/Python)
  2. Low level programming language – Similar to machine code (Assembly/C)

There are thousands of languages. As a programmer one should explore as much as he can to find what is best suited for him and stick with it to become master of that programming language.

Where to write and run your program?

One cannot write the program somewhere and ask your compiler to compile and then execute your program.

There is place or an environment, where one should write their program.

That is where IDE comes into picture

IDE:

IDE stands for integrated development environment

It is a software program which allows us to write program on top of it.

It provides us with a beautiful GUI(Graphical user interface) which makes our life a lot more easier.

Some IDE will also have their own terminal integrated with it. So we don't have to go to the command line to compile and execute our program.

Example: Visual Stduio Code, Eclipse, IntelliJ, Netbeans and so on.

These IDE not only provides us with the environment but all includes,

Error-Checking -> If there is any error, it will show a red underline below that particular piece of text.

Auto-Fill -> It will give us suggestions about the words we want to type or automatically complete the word.

Project-Hierarchy -> We can easily organize and manipulate our files and folders inside the IDE.

IDE consist of:

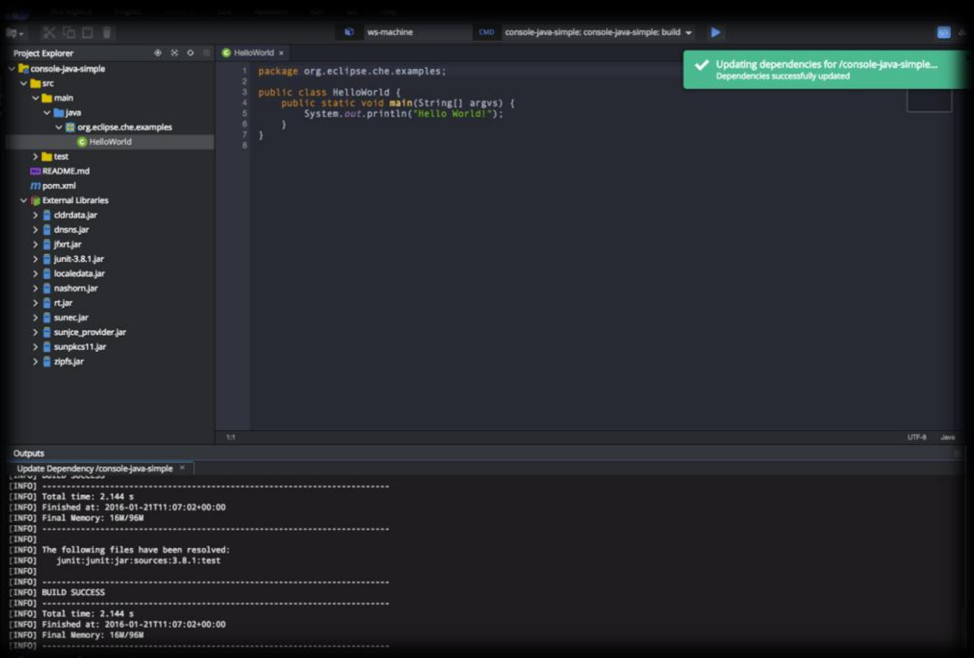
IDE consist of a central area where we will type our program.

There will be project explorer which is used to view our project structure.

There will be tool bar on the top to run, debug or change the appearance of the IDE as per our preferences

It also have a console on the bottom to view our output

The below given image is how an IDE looks. This is Eclipse IDE.



Updating dependencies for /console-java-simple...
Dependencies successfully updated

```
package org.eclipse.che.examples;

public class HelloWorld {
    public static void main(String[] argvs) {
        System.out.println("Hello World!");
    }
}
```

```
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.144 s
[INFO] Finished at: 2016-01-21T11:07:02+00:00
[INFO] Final Memory: 16M/96M
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] The following files have been resolved:
[INFO]    junit:junit:jar:sources:3.8.1:test
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.144 s
[INFO] Finished at: 2016-01-21T11:07:02+00:00
[INFO] Final Memory: 16M/96M
[INFO] ------------------------------------------------------------------------
```

How we should write code inside of an IDE?

We shouldn’t write rubbish inside of the IDE and ask the compiler to compile.

We should write in any of the programming language and that language has it own syntax and as a programmer you should obey the syntax inorder to run your program successfully.

Just like any language have their own grammar. All programming languages have their own syntax and semantics that programmer must follow.

Syntax:

Rules you must follow if you want your program to execute correctly

Syntax for each programming language will be unique

Breaking programming rules will result in an error

Example:

In java, whenever we declare or initialize a variable we should mention the type of the variable before that. Because java is strongly typed and it will check the types during the compile time itself and won’t allow you to compile your program unless and until you are free from error.

Compilation Successful ✔	Compilation Error ✕
int x = 10	x = 10

However in javascript, it won’t cause any error

Real-Life-Example:

“Let’s Eat, Grandma ! -> Calling grandma to eat with you

“Let’s Eat Grandma ! -> Saying that you want to eat grandma

How do we get information from the computer?

Programmers usually keep track of their progress by looking at the console.

The console will display the output for your program.

If you wrote a program which tells the computer to print the result on to the console. Then, the console will display the result on the console.

But if you didn’t used the method to print the result on the screen. It will perform the operations but however you won’t will see nothing in the console.

Example:

Test.java

```
Class Test {

Public static void main(String[] args){

int x = 10;

int y = 20;

int result = x + y;

System.out.println(“The Result Is : “ + result);

}

}
```

Console

```
The Result Is 30.
```

- The Console is mainly a developer tool. It is not meant to be interacted by the end user.
- It should be used by the end only if it is a text-based game/ simple programs.

What can a computer do by itself?

A computer already knows how to do simple arithmetic.

Addition(+), subtraction(-), multiplication(*) and division(/).

Most programming languages has an additional operator known as modulus (%) which gives us the reminder of the division.

Modulus operator has several usecases. To find whether the number is even or odd we can use modulus operator.

Even or odd:

```
Class OddOrEven{

Public static void main(String[] args){

Scanner scanner = new Scanner(System.in)
System.out.println(“Enter the number: “);

int number = scanner.nextInt();

if(number%2==0){

System.out.println(“Even Number”)

}

else{

System.out.println(“Odd Number”);

}

}

}
```

Example:

Assume that you want to build a calculator app

We have to tell the computer through a set of instructions what it should do based on the user input.

It the user clicked on add then it should perform add operation on those operands and give the result back to the user

Strings:

Not only with operator. Computer can also work with the strings.

Strings are nothing but text or a sequence of characters and it should be enclosed within double quotation marks.

We can also concatenate string with another string or integer or anything else.

This is what we call as String Concatenation. Concatination operator(+).

Example:

```
     System.out.println(“This is Deepthi and I am “+18+” years old”);
```

In the above example we are concatenating the string with the integer and again with the string

String + Integer => String & Integer + String => String

Integer + Integer => Integer

“4” → String

4 → integer

Both are not equal

Variables:

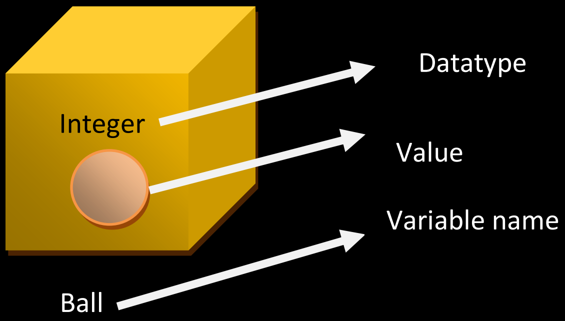
Variable is something that can store information.

We can consider variable as a container which will store a value. The datatype is like a label on the container which mentions what type of values should be stored inside the variable. The actual value is like a content inside the container.

Integer variable stores only integer value.

Each and every value occupies memory inside our computer.

Variable is like a named memory location. So we can reference the value and manipulate it whenever needed.



DataTypes:

There are two types of datatypes:

1. Primitive Datatypes -> Built into that programming language by default and we can use it directly.
   Both the reference and the value of the primitive datatypes will be stored in the stack
2. Reference Datatype -> These are the custom datatypes which we is manually created by the users based on the requirements.
   Here the reference will be stored in the stack and the actual object created using new keyword will be stored in heap.

Primitive Datatypes:

There are 8 types of primitive datatypes.

They are,

1. byte
2. short
3. int
4. long
5. float
6. double
7. boolean
8. char

Non-Primitive Datatypes:

Array

String

List etc…

Why variables are usefull:

You will use a lot of data in your program like username, password etc.

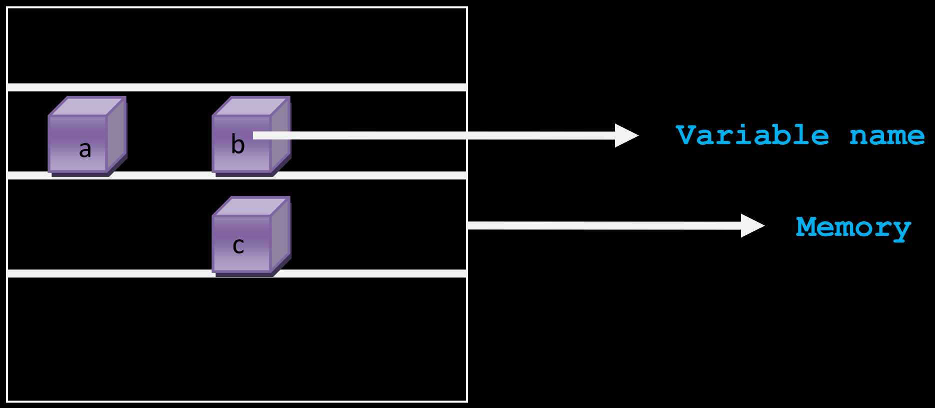
These data’s or values will be stored in some part of your javamemory inside your JVM.

But we have to do certain operations using the data for that we can use the variable to name the data.

So we can reference the data whenever needed

What happens when we create a variable:

Creates a little space in the memory that stores your variable name along with the content.



Dual Pointed Variables:

When two variable names point to the same content. It is said to be dual point variables.

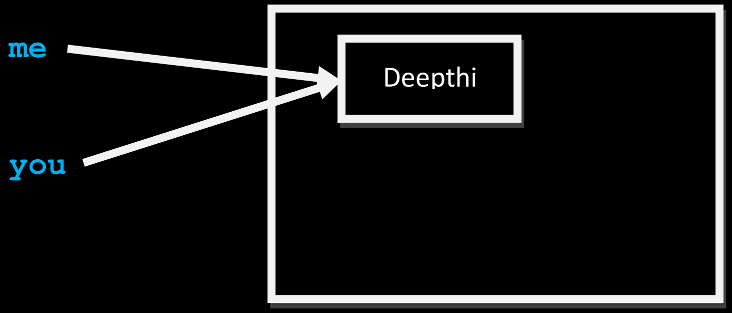
Example:

```
Class DualPointVaribales{

Public static void main(String[] args){

String me = “Deepthi”

String you = “Deepthi”

}

}
```

Here, a variable is created with a value of “Deepthi” and again another variable is created with the same value.

Here separate memory location will not be allocated for both the values to save the memory.

But what happen is the newly created variable name will point to the value of the first variable.


me
you
Deepthi