Community Link:
https://community.simplilearn.com/forums/fsd-jg-may-2023.1551/post-thread

1. What is a program ?

   A program is defined as the set instructions which are intended to perform some activity.

2. What is Programming Language?

   A programming language is defined as the set of instruction used establish a communication between the programmer and the machine.

3. What are the different levels of programming ?

   The Programming world is divided into 3 parts:

   i. Machine Level Language —> 1945's
      - Code is written in 0's and 1's
      - Difficult to debug. (Finding the root cause of the issue)
      - Difficult to understand

   ii. Assembly Level Language —> early 1960's
      - Code is written in Pneumonic format
      - Eg: ADD, SUB, MUL, DIV, STR
      - Debugging is simple
      - Confusion in can case of registers
      - We make use of **Assembler** software to convert ALL-MLL

   iii. High Level Languages —> 1960's
      - Code is written in symbolic fashion
      - Ex: +, -, *, /
      - Very easy to debug compared to any other language
      - Easy understood by humans
      - We make use of **Compiler** software to convert HLL-MLL

Note:
1. The computer can understand only 0's and 1's
2. Any program that we write must be written in MLL as it contains only binary
3. Assembler is a software that converts ALL-MLL
4. Compiler is a software that converts HLL-MLL

4. What are the different HLL's?

   The most important HLL's are:

   i. Basic Combined Programming Language (BCPL)
      -> First popular HLL
      -> It occupy more memory
      -> Because of the above dis-advantage BCPL became expensive language
      -> Non-Potable

   ii. B Language
      -> Popular HLL
      -> It occupy less memory
      -> Not Structured
      -> Non-Portable

   iii. C Programming Language
      -> Popular HLL from decades
      -> It occupy less memory
      -> It is structured
      -> Non-Portable
      -> It was not Object Oriented

   iv. C++ Programming Language
      -> Popular HLL from decades
      -> It occupy less memory
      -> It is Object Oriented
      -> Non-Portable

   v. Java Programming
      -> Popular HLL from decades
      -> It occupy less memory
      -> Its Object oriented
      -> It is portable
      -> It supports internet
      -> It supports enterprise web application development

5. What is a class in java?

   A class is defined as the Blueprint of an object.
   The class doesnot exist in reality.
   It is with the help of the class that we will be defining how an object looks like
   We can have multiple object for a single class.

6. What is an Object?

   An object is defined as the instance of the class.
   An object is the real world entity which has an existence.

We can create multiple objects of a particular class
We cannot create object of 2 class together as a single entity

7. What is Obejct Orientation?
   - Obejct: real world entity
   - Orientation: Perspective
   - Object Orientation: Perspective towards real world entity

8. What are the principle of Object Orientation?
   There are basically 5 principle of Object Object Orientation
   i. The world is a collection of obejct
   ii. Every object is useful and no object is useless
   iii. Every object is under constant interaction.
   iv. Every object belongs to a class and the class is just a blue print
   v. Every object has something and every object does something

9. Create a simple class with 5 properties and 5 behaviours
   Properties ---> has part ----> data-types
   Behviours ---> does part ----> methods

   Properties : name, age, color, cost, breed
Dog / Car
   Behaviours : run, walk eat, sleep, bark

```
class Dog{
    //Properties
    String name;
    int age;
    String color;
    int cost;
    String breed;
    //behaviours
    void run(){
    }
    void walk(){
    }
    void eat(){
    }
    void sleep(){
    }
    void bark(){
    }
```

}

10. What is the process of compilation?

HLL ---> compiler ---> MLL ---> RAM ---> Processor ---> Output

Incase of compiler the code will be converted to MLL in a single go.

11. What is the process of Interpretation?

High level line ---> interprepter---> Binary Line ---> RAM ---> Processor ---> Output

|_____|

Incase of interpreter the code will be converted to MLL line by line.

12. Is java compiled programming language or an interpreted programming language?

java is both compiled as well as interpreted language.

13. What is the syntax of a method?

a) Layman:

```
output_from_method name_of_method (input_to_method)
{
        ----- (body of method)
}
```

b)Technical:

```
return_type name_of_method (parameters)
{
        ----- (body of method)
}
```

Examples:

```
void add(){
        System.out.println("add method");
}

int add(int a, int b){
        return a+b;
}
```

14. What are the different types of methods?

there are 4 types:

NPNR --> No Parameter No Return_Value

example:

```
void add(){
        System.out.println("add method");
}
```

NPWR --> No Parameter With Return_Value

```
void add(){
        return 10+20;
}
```

WPNR --> With Parameter No Return_Value

```
int add(int a, int b){
        int c = a+b;
}
```

WPWR --> With Parameter With Return_Value

```
int add(int a, int b){
        return a+b;
}
```

15. Hello World program in Java.

```
class Program1{
        public static void main(String args[])
        {
                System.out.println("Hello World");
        }
}
```

Main Method in Java:

i. public : to make it available for all

ii. static : to make it accessible for all and also to execute this method without creating the object.

iii. void : no return value

iv. String args[] : the array of string passed as the input to the main method

Java Code --> Java Compiler --> ILL --> JVM --> MLL --> RAM --> Processor--> Output

ILL --> Intermediate level language code (Byte code)

JVM --> Java Virtual Machine is responsible for converting ILL to MLL

16. What is a variable?

A variable is a storage space where we can store some data.

There are 3 types of variables in Java:

i. instance variables

ii. local variables

iii. static variables

17. What is an Instance variable?

Any variable present directly inside the class is called as the Instance Variable.
The instance variables will get default values allocated from the JVM.
The instance variables will get the memory allocated in the heap segment.
The instance variables are also called as object variables because these variables will always be associated with the object being created.

18. What is an Local variable?

Any variable present inside the method is called as the Local Variable.
The Local variables will not get default values.
The Local variables will get the memory allocated in the stack segment.
The Local variables are not associated with the object being created rather they are associated with the methods inside which they are present.

19. What is an Static variable?

Any variable present directly inside the class and is associated with **"static keyword"** is called as the Static Variable.
The Static vairables are firstly the instance variables with static privilage
The static variables will get default values allocated from the JVM.
The Static variables will get the memory allocated in the static segment.
The Static variables are also called as VIP variables because these variables will get the memory allocated first and only once

20. What are the different types of data in java?

Real world data	Representation in java	format used
textual type of data	char data type	ASCII or UTF
integer type real world data	byte, short, int, long	base-2 format
real number type real world data	float, double	IEEE
image	built in fuctions	jvm dependedent
audio	built in fuctions	jvm dependedent
video	built in fuctions	jvm dependedent
yes/no -- true/false	boolean	jvm dependedent

21. What are the segregations of Datatypes in java?

In Java the datatypes are segregated into 2 types:

i. Primitive Datatype
   ex: int, short, byte ,long, float, double, char, etc
ii. Non-Primitive Datatype
   ex: Objects and Wrapper classes

22. WAP to create atleast 2 object for the class Dog which has atleast 5 properties and 5 behaviours.

    Steps to create Java Project in Eclipse:

    i. Go to File.
    ii. click on new and click Java project.
    iii. Give the name of the project.
    iv. click on finish.
    v. Once the project is created then open the project and delete the module-info.java file if present.

```
package Day2;

class Dog{
//Properties
String name;
int age;
String color;
int cost;
String breed;
//Behaviors
void run(){
System.out.println("Dog runs");
}
void walk(){
System.out.println("Dog walks");
}
void eat(){
System.out.println("Dog eats");
}
void sleep(){
System.out.println("Dog sleeps");
}
void bark(){
System.out.println("Dog barks");
```

```
}
}

public class Program2 {
public static void main(String[] args)
{
//creating Dog object - 1
// 1. when we say new JVM will be
activate and it will ask for the
class for which
// object must be created
// 2. Every object must be handled
using some variables which is of the
same class type
Dog d1 = new Dog();
System.out.println(d1.name);
System.out.println(d1.color);
System.out.println(d1.cost);
System.out.println(d1.age);
System.out.println(d1.breed);
d1.name="tommy";
d1.age=4;
d1.color="white";
d1.cost=5000;
d1.breed="GR";
System.out.println("-------------");
System.out.println("*** Properties
Execution ***");
System.out.println(d1.name);
System.out.println(d1.color);
```

```
System.out.println(d1.cost);
System.out.println(d1.age);
System.out.println(d1.breed);
System.out.println("*** Behaviour
Execution ***");
d1.run();
d1.bark();
d1.eat();
d1.sleep();
d1.walk();
System.out.println("-----------------
--------------");
System.out.println("End of object 1");
System.out.println("-----------------
--------------");
//creating Dog object - 2
Dog d2 = new Dog();
d2.name="jimmy";
d2.age=8;
d2.color="brown";
d2.cost=3000;
d2.breed="pug";
System.out.println("--------------");
System.out.println("*** Properties
Execution ***");
System.out.println(d2.name);
System.out.println(d2.color);
System.out.println(d2.cost);
System.out.println(d2.age);
```

```
System.out.println(d2.breed);
System.out.println("*** Behaviour
Execution ***");
d2.run();
d2.bark();
d2.eat();
d2.sleep();
d2.walk();
}
}
```

23. WAP to print the message “Welcome to SimpliLearn” 5 times using:
    a. for loop
    b. while loop
    c. do-while loop
24.