Assignment

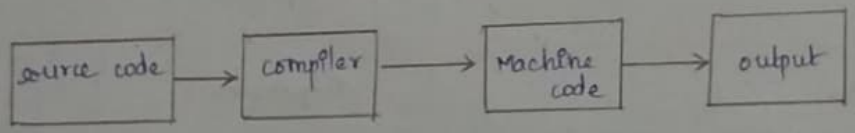
Compiler and Interpretor:

Compiler:

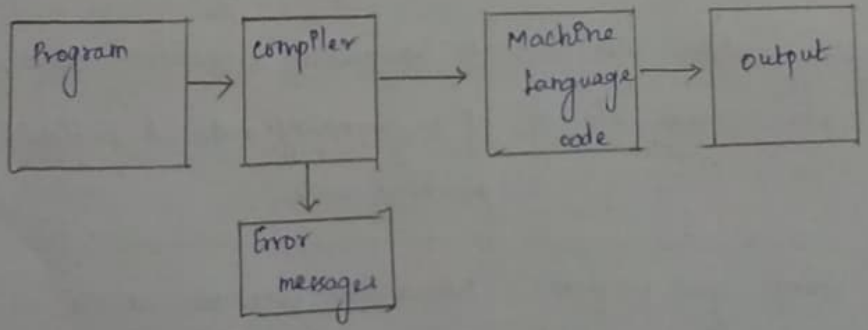
A compiler is a software program that follows the syntax rule of programming language to convert a source code to machine code. It cannot fix any error if present in a program, it generates an error message, and you have to correct it yourself in the program's syntax. If your written program is correct, then the compiler will convert your entire source code into machine code. A compiler converts complete source code into machine code at once. And finally, your program get executed.

How compiler works.

a)



b)

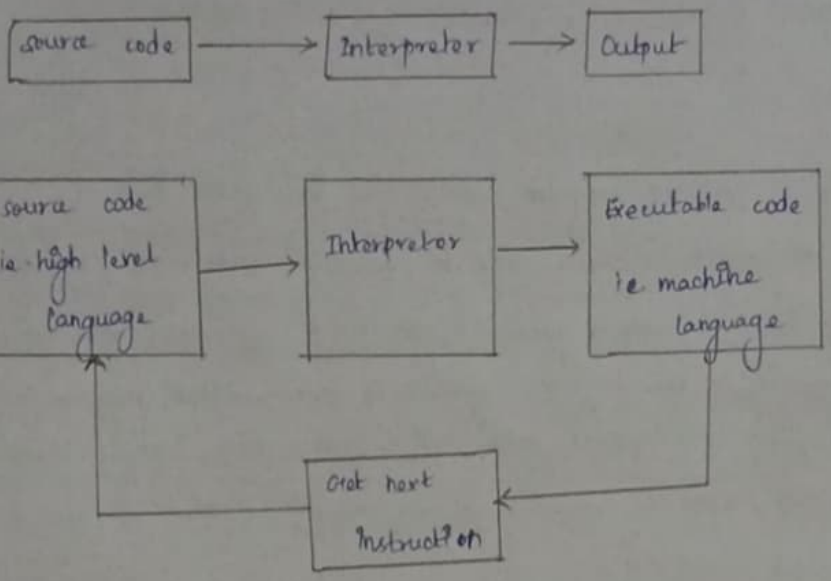


Interpreter:

An interpreter is also a software program that translates a source code into a machine language. However, an interpreter converts high-level programming language into machine language line-by-line while interpreting and running the program.

How Interpreter works

a)



Difference between compiler and Interpreter:

Compiler	Interpreter
* A compiler translates complete high-level programming code into machine code at once.	An interpreter translates one statement of programming code at a time into machine code.
* If you want to change your program for any reason, either by error or logical changes, you can do it only by going back to your source code.	Interpreted programs can run on only those computers which have the same interpreter.
* It stores the converted machine code from your source code program on the disk.	It never stores the machine code at all on the disk.

* A compiler takes an enormous time to analyze source code. However, overall compiled programming code runs faster as comparison to an interpreter.	An interpreter takes less time to analyze source code as compared to a compiler. However, overall interpreted programming code runs slower as comparison to the compiler.
* The compiler generates an output of a program [the form of an exe file] that can run separately from the source code program.	The interpreter doesn't generate a separate machine code as an output program. So it checks the source code every time during the execution.
* The process of program execution takes place separately from its compilation process. Program execution only takes place after the complete program is compiled.	The process of program execution is a part of interpretation steps, so it is done line by line simultaneously.
* A compiled program is generated into an intermediate object code, and it further required linking. So there is a requirement for more memory.	An interpreted program does not generate an intermediate code. So there is no requirement for extra memory.

```java
import java.util.Scanner;

public class KboatAvgMarks
{
    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);
        System.out.print("Enter number of students: ");
        int n = in.nextInt();

        int rollNo[] = new int[n];
        String name[] = new String[n];
        int s1[] = new int[n];
        int s2[] = new int[n];
        int s3[] = new int[n];
        double avg[] = new double[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter student " + (i+1) + " details:");
            System.out.print("Roll No: ");
            rollNo[i] = in.nextInt();
            in.nextLine();
            System.out.print("Name: ");
            name[i] = in.nextLine();
            System.out.print("Subject 1 Marks: ");
            s1[i] = in.nextInt();
            System.out.print("Subject 2 Marks: ");
            s2[i] = in.nextInt();
            System.out.print("Subject 3 Marks: ");
            s3[i] = in.nextInt();
            avg[i] = (s1[i] + s2[i] + s3[i]) / 3.0;
        }

        System.out.println("Roll No\tName\tRemark");
        for (int i = 0; i < n; i++) {
            String remark;
            if (avg[i] < 40)
                remark = "Poor";
            else if (avg[i] < 60)
                remark = "Pass";
            else if (avg[i] < 75)
                remark = "First Class";
            else if (avg[i] < 85)
                remark = "Distinction";
            else
                remark = "Excellent";
            System.out.println(rollNo[i] + "\t"
                + name[i] + "\t"
                + remark);
        }
```