# 1)Execute program from notepad and Eclipse IDE

class Employee{

public static void main (String[] args)

{

System.out.println("hello Vijay");

}}

# **Understand the importance of compiler and JVM**

Java compiler executes the code by step by step process or from top to bottom.

Java compiler first compiles our Java code to bytecode to get understand by the computer.

When the programmer is set to execute the code by entering the file name in the command promt compiler checks wheather the source file is available or not if the file is found it generate the byte code or error if found.

Programmer fixes the errors and runs the compiler again to recompile the program

It generates byte code only if the code is error free or else error is highlighted.

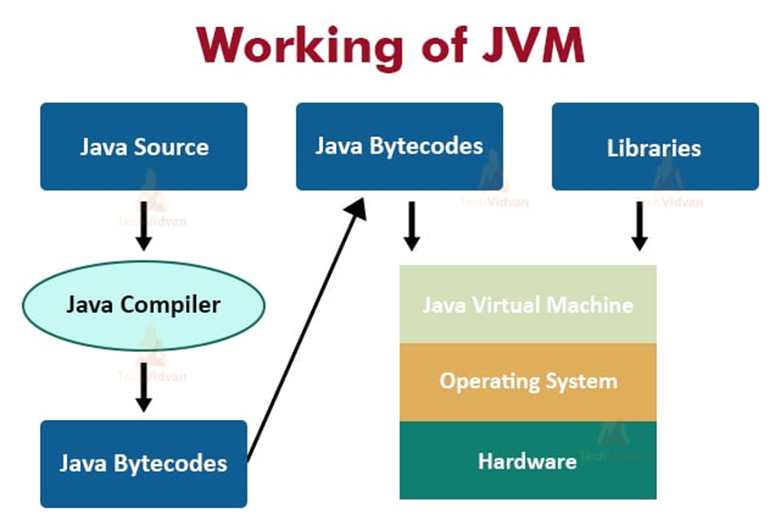
**Java compiler options**

The javac compiler options for programmers. These include:

* **-version.** It displays information about the compiler.
* **-help.** It prints a summary of standard options.
* **-nowarn.** It turns off warnings.
* **-g.** It generate debugging information (line number, source file information, local variables).
* **-verbose.** It generates "verbose" output, including information about compiled source files and loaded classes.

All the above mentioned compiler options are case sensitive.

**JVM**



From the above image we can easily get an idea about the JVM .

The term virtual machine means.

Virtual:Virtual machine is something that is not in the reality but still it exists.

Machine: which makes our works easier.

The JVM operates on specific types of data as specified in Java Virtual Machine specifications. The data types can be dividedinto primitive types (integers Floating-point, long etc.) and Reference types. The earlier JVM were only 32bit machines. long and double types.

**KEYWORDS**

**Java Keywords and Their Uses**

* **abstract**: This keyword is used to declare a class that cannot be instantiated on its own and must be subclassed. It can also be used to declare methods that do not have an implementation and must be implemented in a subclass.
* **assert**: It is used for debugging purposes, allowing developers to test assumptions in the code.
* **boolean**: This data type keyword is used to declare variables that can only hold two possible values: true or false.
* **break**: It is used to exit from a loop or a switch statement prematurely.
* **byte**: This data type keyword is used to declare variables that can hold an 8-bit integer.
* **case**: It is used within a switch statement to mark blocks of text.
* **catch**: This keyword is used to catch exceptions generated by try statements.
* **char**: It is used to declare variables that can hold a single 16-bit Unicode character.
* **class**: This keyword is used to define a new class.
* **continue**: It is used to skip the current iteration of a loop and continue with the next iteration.
* **default**: In a switch statement, it specifies the default block of code that will execute if no case matches.
* **do**: It is used to create a do-while loop, which will execute a block of code at least once before checking a condition.
* **double**: This data type keyword is used to declare variables that can hold a double-precision 64-bit floating-point number.
* **else**: It is used to specify a block of code that will execute if the condition in an if statement is false.
* **enum**: This keyword is used to declare an enumerated type, a special data type that enables variable to be a set of predefined constants.
* **extends**: It is used to indicate that a class is derived from another class.
* **final**: This keyword is used to declare constants or methods that cannot be overridden or classes that cannot be subclassed.
* **finally**: It is used to create a block of code that will execute after a try-catch block, regardless of whether an exception was thrown.
* **float**: This data type keyword is used to declare variables that can hold a single-precision 32-bit floating-point number.
* **for**: It is used to create a for loop, which allows you to execute a block of code multiple times with different values.
* **if**: This keyword is used to create a conditional statement that will execute a block of code if a specified condition is true.
* **implements**: It is used when a class wants to implement an interface.
* **import**: This keyword is used to bring a package, class, or interface into visibility.
* **instanceof**: It is used to test whether an object is an instance of a specific class or interface.
* **int**: This data type keyword is used to declare variables that can hold a 32-bit signed integer.
* **interface**: It is used to declare a special type of class that can only contain abstract methods.
* **long**: This data type keyword is used to declare variables that can hold a 64-bit integer.
* **new**: It is used to create new instances of objects.
* **package**: This keyword is used to declare a package that groups related classes and interfaces.
* **private**: It is an access modifier that restricts visibility to the same class only.
* **protected**: This access modifier keyword allows visibility within the same package or subclasses.
* **public**: It is an access modifier that allows visibility from any other class.
* **return**: This keyword is used to exit a method and optionally pass back a value to the caller.
* **short**: It is used to declare variables that can hold a 16-bit integer.
* **static**: This keyword is used to declare class-level variables or methods that can be accessed without needing an instance of the class.
* **super**: It is used to refer to the immediate parent class of the current class.
* **switch**: This keyword is used to select one of many code blocks to be executed.
* **synchronized**: It is used to indicate that a method can only be accessed by one thread at a time.
* **this**: This keyword is used to refer to the current instance of a class.
* **throw**: It is used to explicitly throw an exception.
* **throws**: This keyword is used to declare the exceptions that a method might throw.
* **transient**: It is used to indicate that a field should not be serialized.
* **try**: This keyword is used to define a block of code in which exceptions will be handled.
* **void**: It is used to declare that a method does not return any value.
* **volatile**: This keyword is used to indicate that a variable's value will be modified by different threads.
* **while**: It is used to create a while loop, which executes a block of code as long as a specified condition is true.

**Java Control Statements | Control Flow in Java**

Java compiler executes the code from top to bottom. The statements in the code are executed according to the order in which they appear. However,java provides statements that can be used to control the flow of Java code. Such statements are called control flow statements. It is one of the fundamental features of Java, which provides a smooth flow of program.

Java provides three types of control flow statements.

1. Decision Making statements
   * if statements
   * switch statement
2. Loop statements
   * do while loop
   * while loop
   * for loop
   * for-each loop
3. Jump statements
   * break statement
   * continue statement

Decision-Making statements:

As the name suggests, decision-making statements decide which statement to execute and when. Decision-making statements evaluate the Boolean expression and control the program flow depending upon the result of the condition provided. There are two types of decision-making statements in Java, i.e., If statement and switch statement.

In jaava are similar to if-else-if statements. The switch statement contains multiple blocks of code called cases and a single case is executed based on the variable which is being switched. The switch statement is easier to use instead of if-else-if statements. It also enhances the readability of the program.