The high level program is known as source program and the corresponding machine level known as object program

Compilers and interpreters are used to convert the code of high level language into machine language.

Assembler is used for converting the code of low level language (assembly language) into machine level language.

**What is compiler?**

A compiler searches all the errors of a program and lists them.

If the program is error free then it converts the code of program into machine code and then the program can be executed by separate commands.

Compiler converts source code to some kind of intermediate form.

For static language (*that language in which we also specify the code entered*) , a compiler usually converts the source code to assembly, which usually did not get stored to disk, then the assembler convert the assembly to binary code, which is usually stored as object file(.o or .obj suffix usually), then linker is invoked(*appeal for confirmation*) to link object file(s) to binary executable. Also it is common to refer to this whole process of compiling, assembling, linking as compiling.

**What is an Interpreter?**

Interpreters are tools that execute instructions written in some programming language.

Interpreter can either directly execute high level source code or translate them to intermediate code and then interpret it or execute precompiled code.   
An interpreter checks the errors of a program statement by statement. After checking one statement, it converts that statement into machine code and then executes that statement. The process continues until the last statement of program occurs.

Languages like Perl, Python, MATLAB and Ruby are examples of programming languages that use an intermediate code. UCSD Pascal interprets a precompiled code. Languages like Java, BASIC and Samlltalk first compile the source to an intermediate code called bytecode and then interpret it.

**What is an Assembler?**  
Assembler is used for converting the code of low level language (assembly language) into machine level language.  
Assembler is software or a tool that translates Assembly language to machine code. So, an assembler is a type of a compiler and the source code is written in Assembly language. Assembly is a human readable language but it typically has a one to one relationship with the corresponding machine code. Therefore an assembler is said to perform isomorphic (one to one mapping) translation. Advanced assemblers provide additional features that support program development and debugging(a process of locating and removing bugs,errors or abnormalities) processes. For example, the type of assemblers called macro assemblers provides a macro facility. **What is the difference between an Assembler and an Interpreter?**  
An assembler can be considered a special type of compiler, which only translates Assembly language to machine code. Interpreters are tools that execute instruction written in some language. Interpreter systems may include a compiler to pre-compile code before interpretation, but an interpreter cannot be called a special type of a compiler. Assemblers produce an object code, which might have to be linked using linker programs in order to run on a machine, but most interpreters can complete the execution of a program by themselves. An assembler will typically do a one to one translation, but this is not true for most interpreters. Because Assembly language has a one to one mapping with machine code, an assembler may be used for producing code that runs very efficiently for occasions in which performance is very important (for e.g. graphics engines, embedded systems with limited hardware resources compared to a personal computer like microwaves, washing machines, etc.). On the other hand, interpreters are used when you need high portability. For example, the same Java bytecode can be run on different