Date — / INTEGRATED DEVELOPMENT ENVIORMENT (IDE)

Integrated development environment (IDE):- An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development.

An IDE normally consists of a source code editor, build automation tools, and a debugger. Most modern IDEs contain a compiler, interpreter or both. The boundary between an integrated development environment (IDE) and other parts of the broader software development is not well defined. Sometimes a version control system or various tools system simplify the construction of a graphical user interface (GUI), are integrated. Many modern IDEs also have a class browser, an object browser, and a class hierarchy diagram for use in object oriented software development.

Discussion

High-level language program are usually written (coded) or ASCII text into a source code file.

For ex. .asm.c.cpp.java.js.py is used to identify or a source code file.

In Integrated development Environment or IDE, program written in a high-level languages are either directly executed by some kind of interpreter or converted into machine code by a compiler.

For CPU to execute JavaScript, perl, Python, and Ruby are example of integrated program

Date __/__/____

interpreted programming languages.
C, C++, C#, Java and Swift are examples of compiled programming languages.
The following figure shows the process of activity in an IDE as a programmer enters the source code and then directs the IDE to compile and run the program.

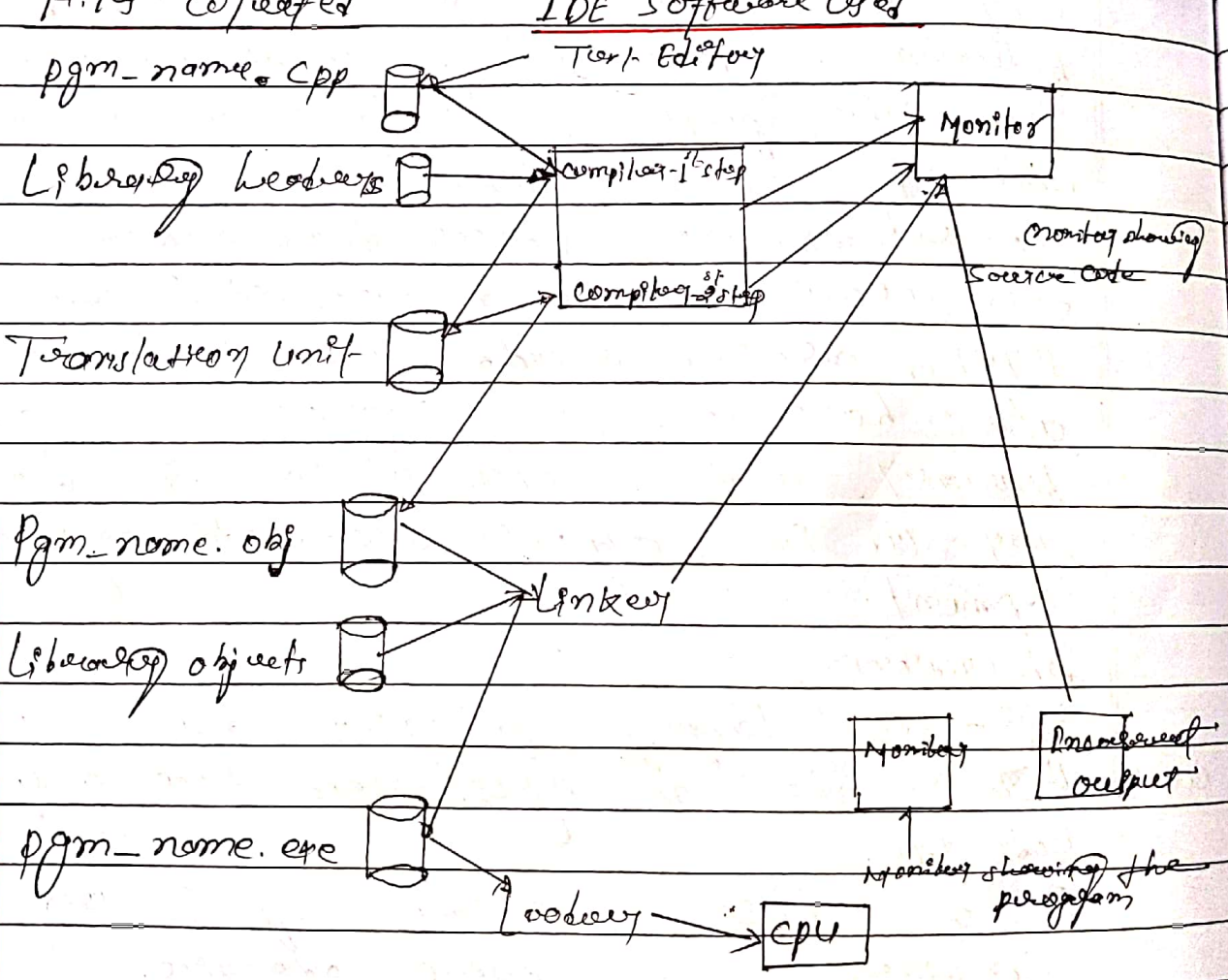

Fig: Integrated Development Environment or (IDE)

When we start the IDE software the programmer usually indicates the file that he want to open for editing or source.
The IDE does the following steps.

1) If there are any unsaved changes to

Date __/__/____

to the source code file it has the text editor save the changes.

II) The compiler opens the source code file and does the first step. which is executing the pre-processor compiler directives and other steps needed to get the file ready for the second step. The #include will insert the header files into the code at this point. If it encounter error, it stops the process and returns to user to the source code file within the text editor with an error message. If there is no problem encountered it save the source code to a temporary file called a translation unit.

III) The compiler opens the translation unit file and does the second step which is converting the programming language code to machine instructions for the CPU, a data area, and a list of items to be resolved by the linker. If there is any problem encountered (usually a syntax or violation of the programming language rules) stops the process and returns to the user to source code file within the text editor with an error message. If no problems encountered it saves the machine instructions, data area, and linker resolution list to an object file.

Page No.

Date __/__/____

IV) The linker opens the program object file and links it with the library objects files as needed. Unless all linker items are resolved, the process stops and returns the user to the source code file within the text editor with an error message. If no problems encountered it saves the linked objects as an executable file.

V) The IDE directs the operating system's program called loader to the executable file into the computer's memory and have the CPU starts processing the instructions. As the user interacts with the program, entering the test data he or she might discover that the outputs are not correct. These type of errors called logic errors and would require the user to return to the source code to change the algorithm.

Types of errors

There are three types exist in a program

i) Compiler errors.
ii) Linker errors.
iii) Logic errors.