Compiler

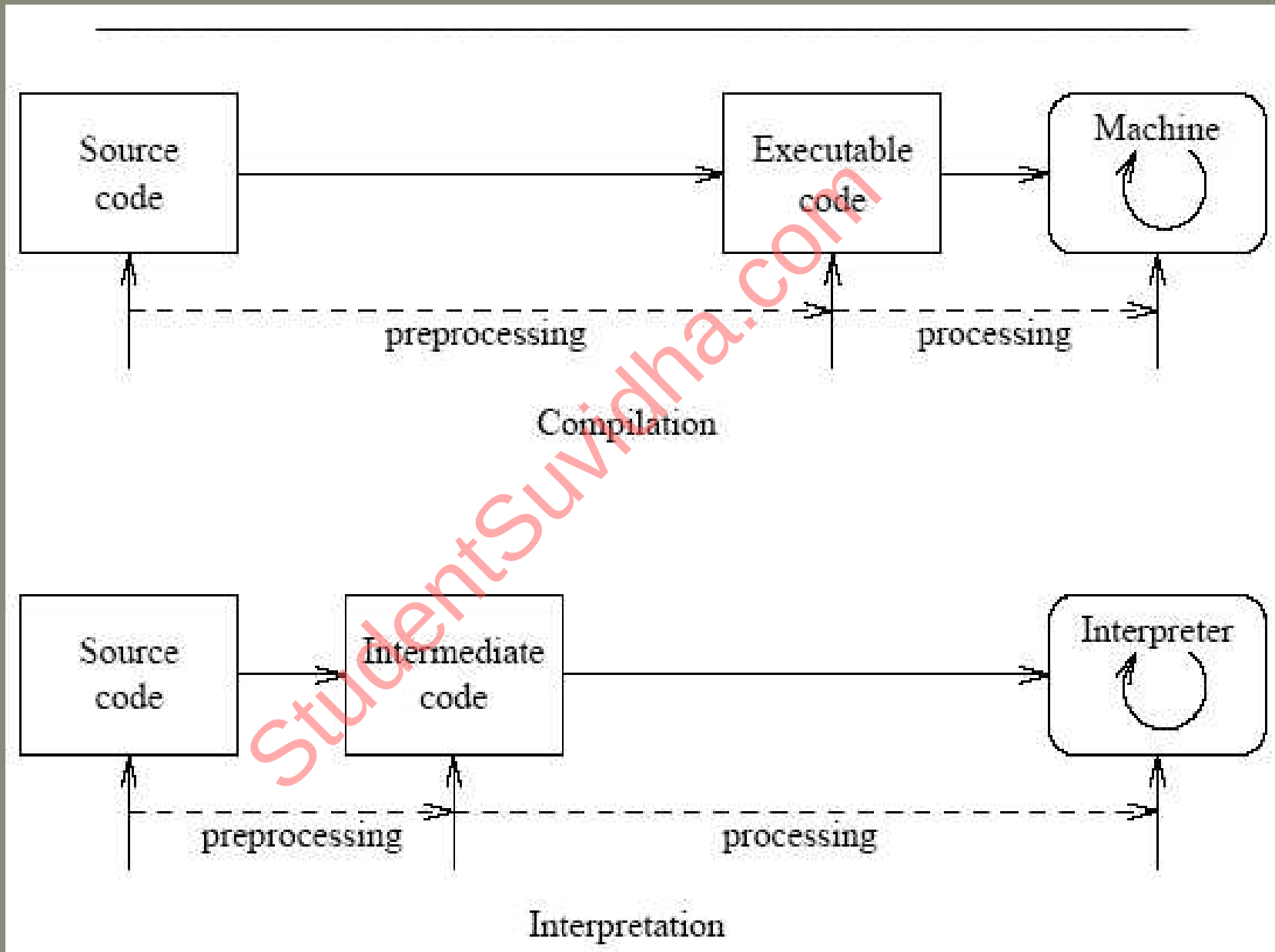
- Compiler: A compiler is program that converts the instruction of a high level language into machine language as a whole. A program written in high level language is called source program. After the source program is converted into machine language by the compiler, it is called an object program.
- The compiler checks each statement in the source program and generates machine instructions. Compiler also checks syntax errors in the program. A source program containing an error cannot be compiled into an object program.
- A compiler can translate the programs of only that language for which it is written. For example C++ compiler can translate only those programs, which are written in C++. Each machine required a separate compiler for each high level language.

Interpreter

- Interpreters: An interpreter is a program that converts one statement of a program at a time. It executes this statement before translating the next statement of the source program. If there is an error in the statement, the interpreter will stop working and displays an error message.

  The advantage of interpreters over compilers is that an error is found immediately. So the programmer can make corrections during program development.

  The disadvantage of interpreter is that it is not very efficient. The interpreter does not produce an object program. It must convert the program each time it is executed. Visual basic uses interpreter.



Compiler characteristics:

- spends a lot of time analyzing and processing the program
- the resulting executable is some form of machine- specific binary code
- the computer hardware interprets (executes) the resulting code
- program execution is fast

Interpreter characteristics:

- translates source code into some efficient intermediate representation (code) and immediately executes this
- relatively little time is spent analyzing and processing the program
- the resulting code is some sort of intermediate code
- the resulting code is interpreted by another program
- program execution is relatively slow

Compiler vs. Interpreter

Compiler

- Pros
  - Less space
  - Fast execution
- Cons
  - Slow processing
    - Partly Solved (Separate compilation)
  - Debugging
    - Improved thru IDEs

Interpreter

- Pros
  - Easy debugging
  - Fast Development
- Cons
  - Not for large projects
    - Exceptions: Perl, Python
  - Requires more space
  - Slower execution
    - Interpreter in memory all the time

LOADER

- Loader is an operating system utility that copies programs from a storage device to main memory, where they can be executed.

Types of loader:

- Absolute Loader: is a primitive type of loader which does only the loading function. It loads the object program from translation time address & simply transfers the control to it. It does not perform linking & program relocation.
- Bootstrap loader: when the computer is turned on bootstrap loader is executed. This loader is responsible for loading the operating system & transferring control to it. The boot strap loader is present in the ROM area of main memory.
- Re-locatable loader: This loader is responsible for relocation & loading.

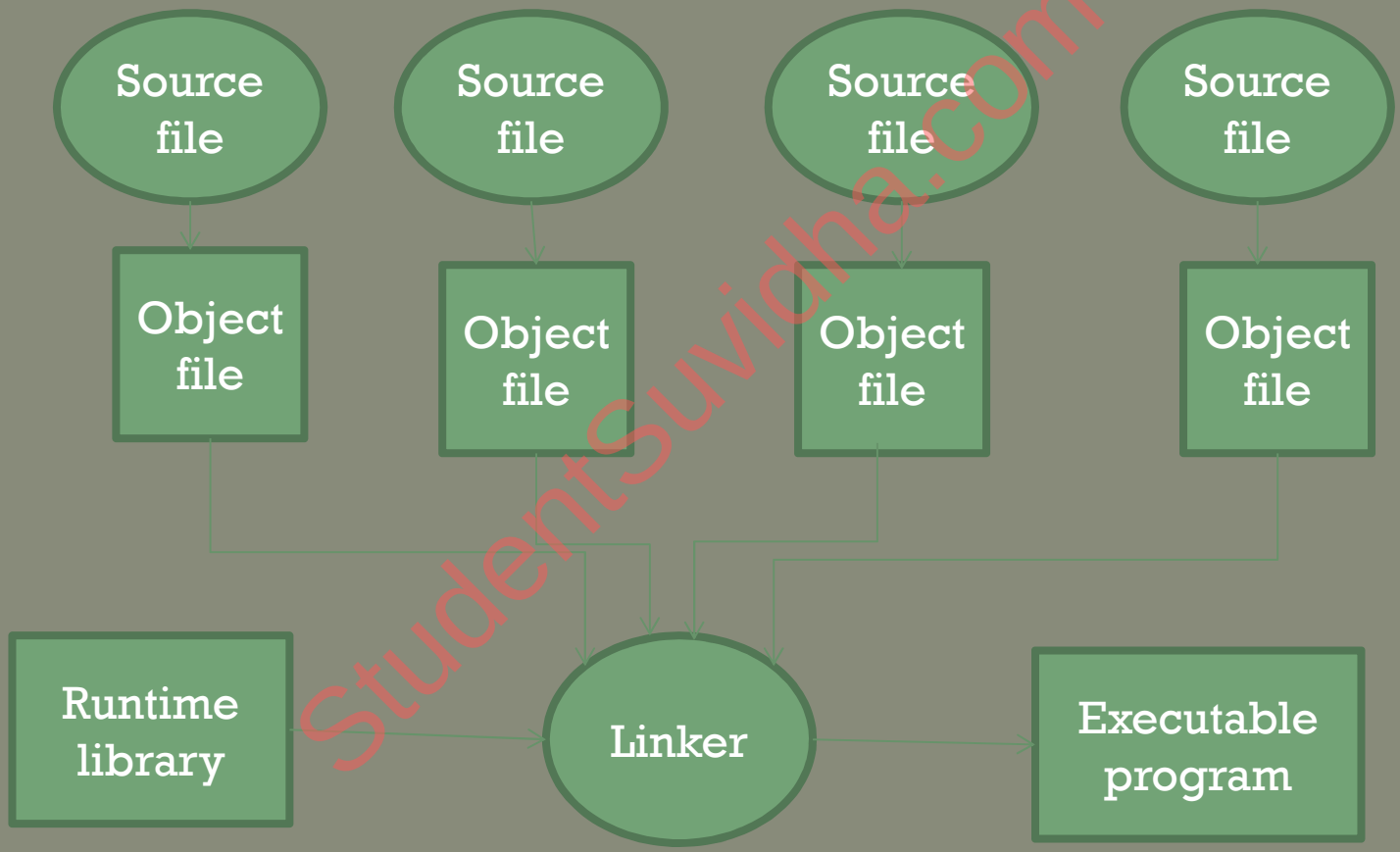
LINKER

- A program that combines object modules to form an executable program. It is also called link editor or binder.
- Many programming languages allows to write different pieces of code, called modules, separately. This simplifies the programming task because you can break a large program into small, more manageable pieces. Eventually, though, you need to put all the modules together. This is the job of the linker to integrate all the modules.
- Linking is the process of combining various pieces of code & data together to form a single executable file that can be loaded in memory.
- Linking can be done at compile time, at load time(by loaders), & also at run time (by application programs).

Types of Linker

- Linking Loader: It performs all the linking & relocation operations and loads the linked program directly into the main memory for execution.
- Dynamic linker: This scheme postpones the linking function until execution time. Any subroutine is loaded and linked to the rest of the program when it is first called.

LINKER (Contd…)



ASSIGNMENT

- Differentiate between:

Compiler and Interpreter

Linker and Loader