

Name : Abdullah Bin Noman Tabid

Student ID : 2120180081

Course Code : CSI-411

Course Title : Compiten

"Final Assessment"

Ans-to-the-Q. No-1

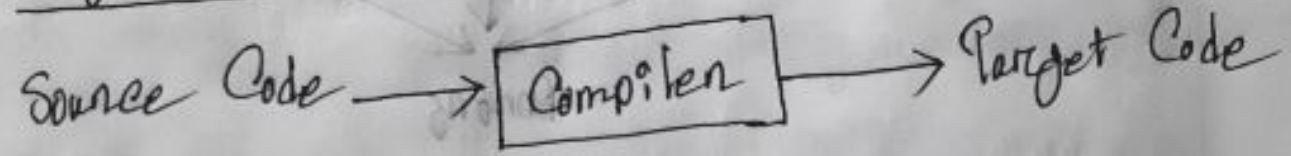
(a)

Types of Compiler:

Following are the different types of compiler-

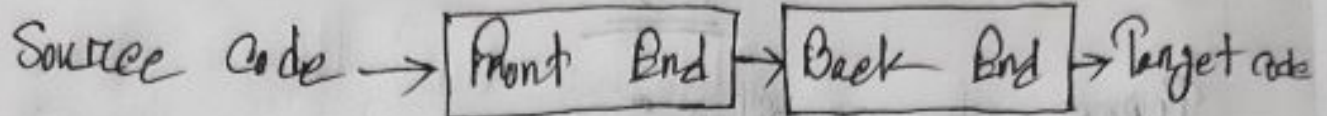
- ① Single Pass Compiler.
- ② Two Pass Compiler.
- ③ Multipass Compiler.

① Single Pass Compiler:



In single pass compiler source code directly transforms into machine code. For example, Pascal language.

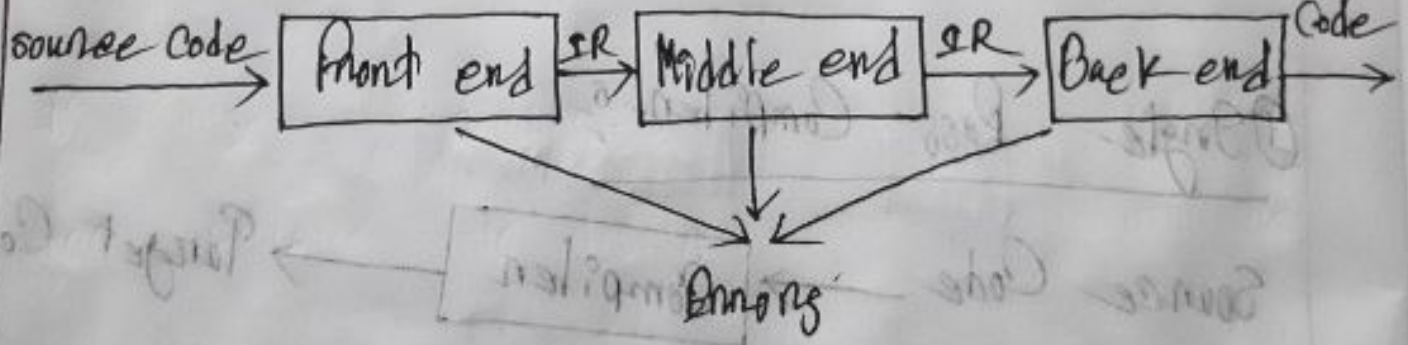
2) Two Pass Compiler;



Two pass compiler is divided into two sections -

- ① Front End.
- ② Back End.

3) Multipass Compilers;



Multipass compiler requires less memory.

It is also known as "Wide Compiler".

3

Phases of compiler:

The typical phases of a compiler are -

① Lexical Analysis: It is also known as scanning.

② Syntax Analysis: It is also known as parsing.

③ Semantic Analysis: This phase checks whether the code is semantically correct.

④ Intermediate Code Generation: This phase generates an intermediate representation of the source code that can be easily translated into machine code.

⑤ Optimization: It improves the performance of the generated machine code.

⑥ Code Generation: It's the final phase.

Ans-to-the Q-No-1(b)Advantages of compiler:

- ① Improved performance.
- ② Portability.
- ③ Increased security.
- ④ Debugging support.
- ⑤ No dependencies.

~~It~~ These advantages are very important for anyone to design any program. The program is fluent and clearly possible to understand.

Disadvantages of compiler:

- ① Compilation time.

② Error detection.

③ Portability.

④ Execution speed.

⑤ Lack of flexibility.

⑥ Resource consumption.

In a compiler designed program there is many errors in the program. Speed is very slow. Virus can be detected very rapidly. Virus can't be destroyed instantly.

Resources used in the compiler program are very costly and less usable.

Resources almost everything are less

flexible.

6

Ans. to the Q No. 1

(c)

Compiler design helps full implementation of High-Level Programming Languages.

Support optimization for Computer

Architecture Parallelism A compiler is a program that translates a high-level language (for example, C, C++, and Java) into a low-level language (object program or machine program).

Java program can be translated into almost all languages of the world.

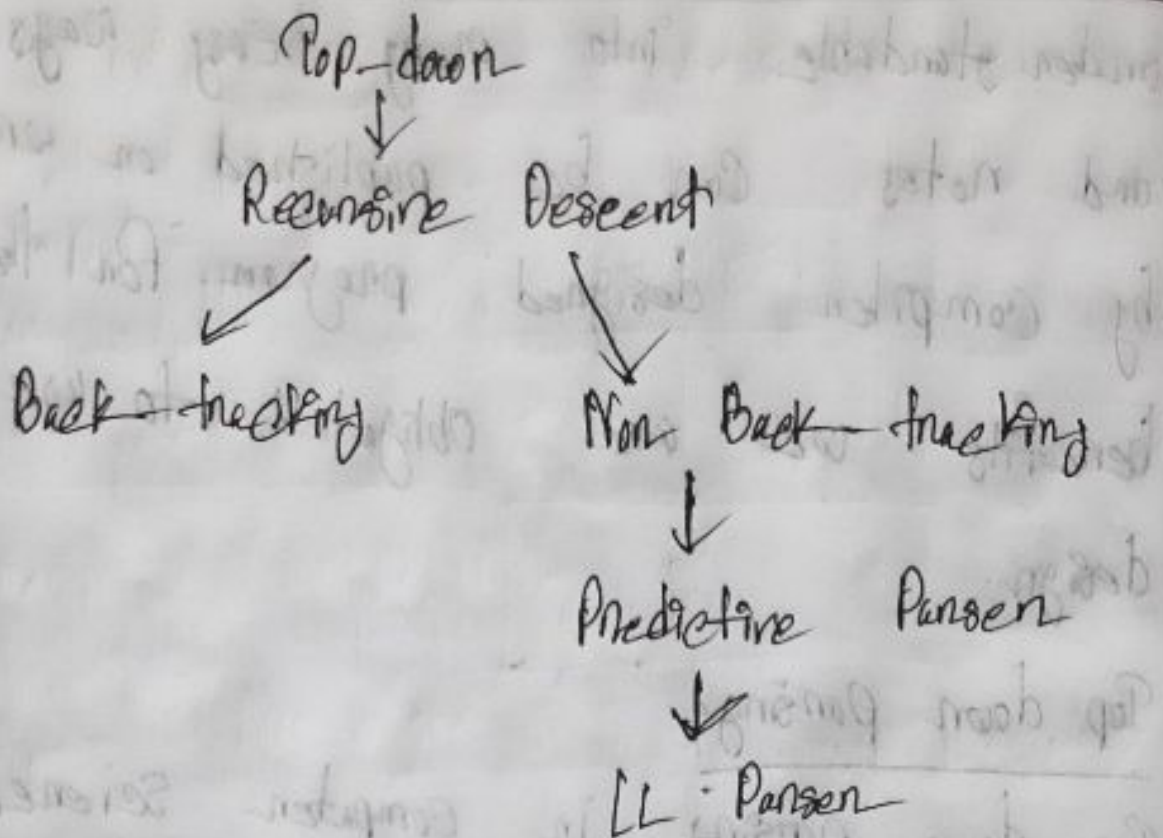
C and C++ programs can also be

understandable into many easy ways. Books and notes can be published or written by computer designed program. For these benefits we are obligated to use compiler design.

Top-down parsing

Top-down parsing in computer science is a parsing strategy when one first looks at the highest level of the parse tree and works down the parse tree by rewriting rules of a formal grammar.

LL parsers are a type of parser that uses a top-down parsing strategy.



Bottom-up parsing;

Bottom-up (or shift-reduce) parsers work by "shifting" symbols onto a stack until the top of the stack contains a right-hand side of a production. The stack is then "reduced"

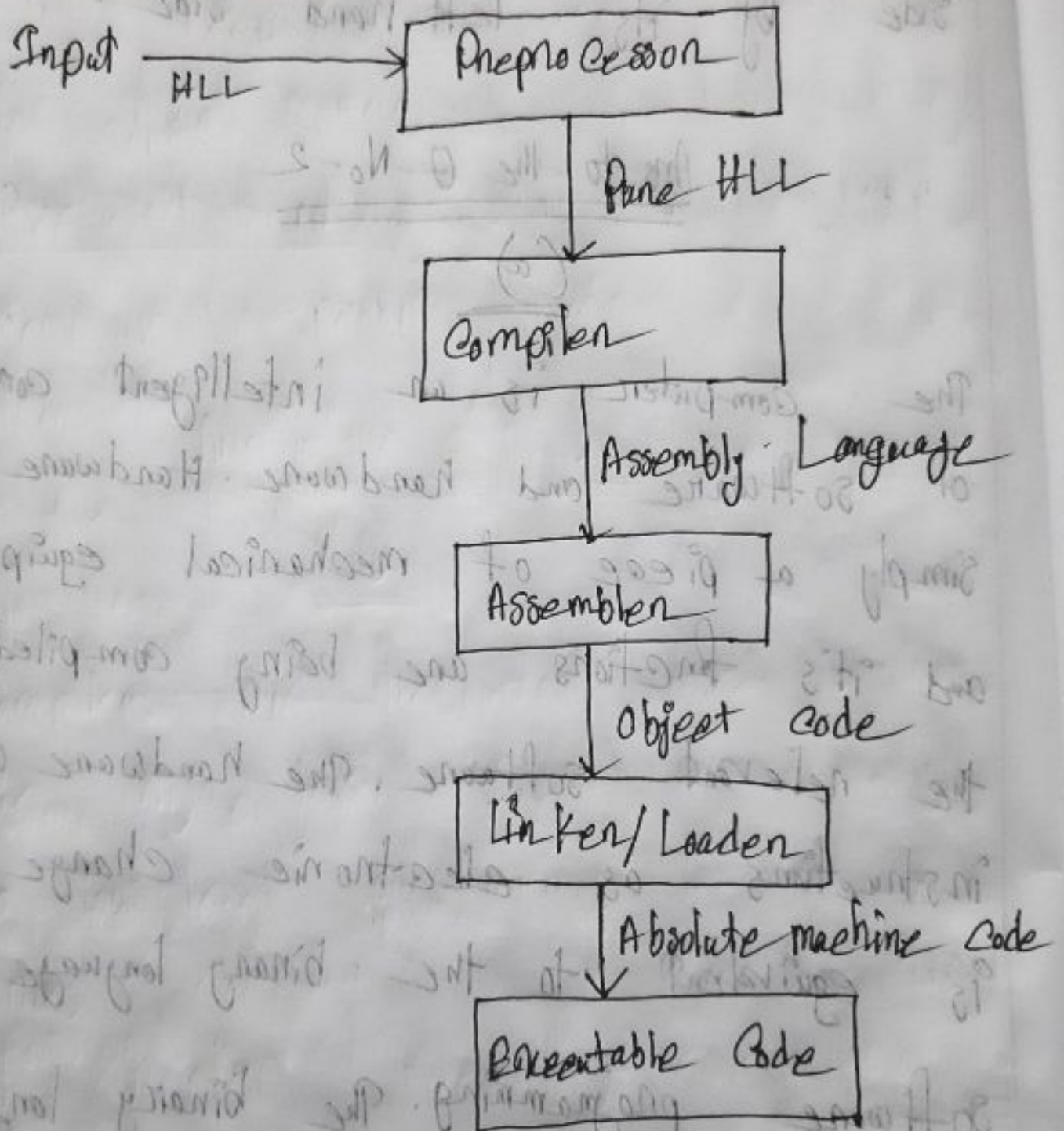
9 81
by replacing the productions right-hand side by its left-hand side.

Ans to the Q-No-2

(a)

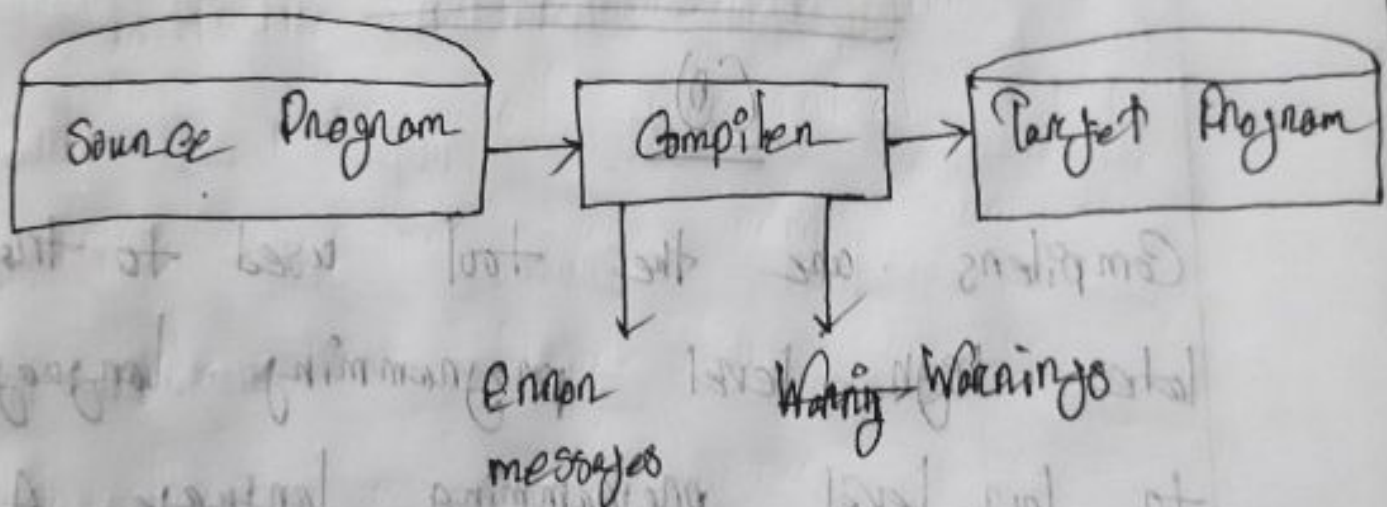
The computer is an intelligent combination of software and hardware. Hardware is simply a piece of mechanical equipment and its functions are being compiled by the relevant software. The hardware considers instructions as electronic charge, which is equivalent to the binary language in software programming. The binary language has only 0s and 1s. This is known

as a language processing system



Preprocessor: It includes all header files and also evaluates whether a macro is included.

Compiler: The compiler takes the modified code as input and produces the target code as output.



Assembler: It produces relocatable machine code as output.

Linker: Takes a collection of objects and combines them into an executable.

Program:
Loader: keeps the linked program in the main memory.

Executable Code: It is low-level and machine specific code.

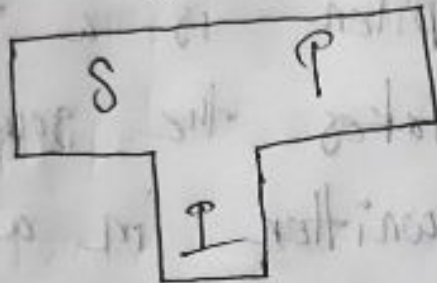
Ans-to-the-Q-No 2



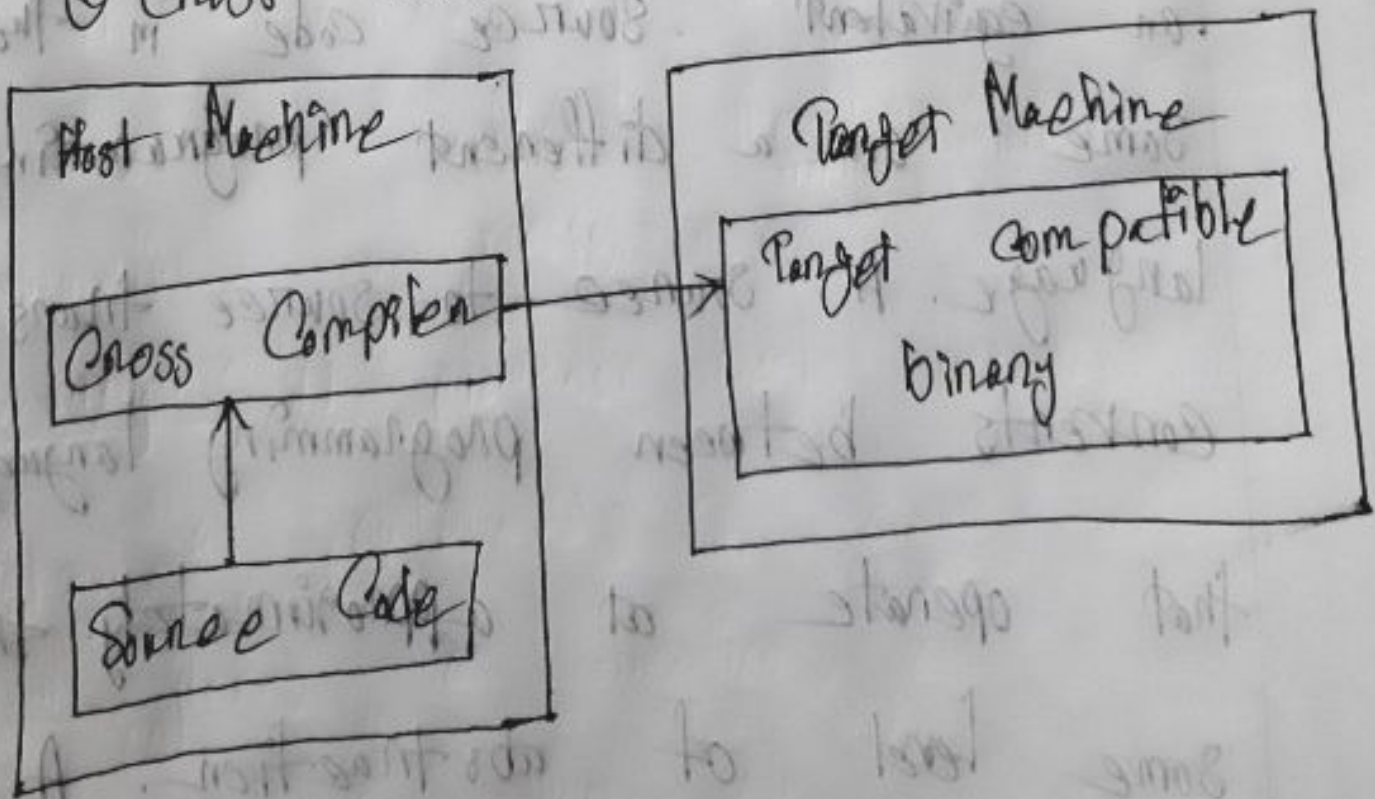
Compilers are the tool used to translate high-level programming language to low-level programming language. A cross compiler is a compiler capable of creating ~~executing~~ executable code for a platform other than the one on which the compiler is running.

For example, a cross compiler executes on machine X and produces machine code for machine Y.

① Diagram for Cross Compiler -



② Cross Compiler Operation -



Ans to the Q No 2

(c)

A source-to-source translator, source-to-source compiler (S2S compiler) or transpiler, is a type of translator that takes the source code of a program written in a programming language as its input and produces an equivalent source code in the same or a different programming language. A source-to-source translator converts between programming languages that operate at approximately the same level of abstraction. A

15

Source-to-source translator may perform a ~~source~~ translation of a program from Python to JavaScript, while a traditional compiler translates from a language like C to assembly or Java to bytecode.