COMPILER
TUTORIAL SHEET - I

Instructor	Email	Office Location & Hours
Dr. Nayyar Ahmed Khan	nayyar@su.edu.sa	CCIT: 8:00 AM – 3:00 PM
Dr. Nouf Altmami	naltamami@su.edu.sa	

Session I		Reading	Exercises
1.	Introduction	Ullman/Aho/Sethi	--

A compiler translates the code written in one language to some other language without changing the meaning of the program.
The high-level language is converted into binary language in various phases. A compiler is a program that converts high-level language to assembly language.
It is also expected that a compiler should make the target code efficient and optimized in terms of time and space.

Compiler design principles provide an in-depth view of translation and optimization process.
Compiler design covers basic translation mechanism and error detection & recovery.
It includes lexical, syntax, and semantic analysis as front end, and code generation and optimization as back-end.

Language Processing System:
We have learnt that any computer system is made of hardware and software.
The hardware understands a language, which humans cannot understand.
We write programs in high-level language, which is easier for us to understand and remember.
These programs are then fed into a series of tools and OS components to get the desired code that can be used by the machine.

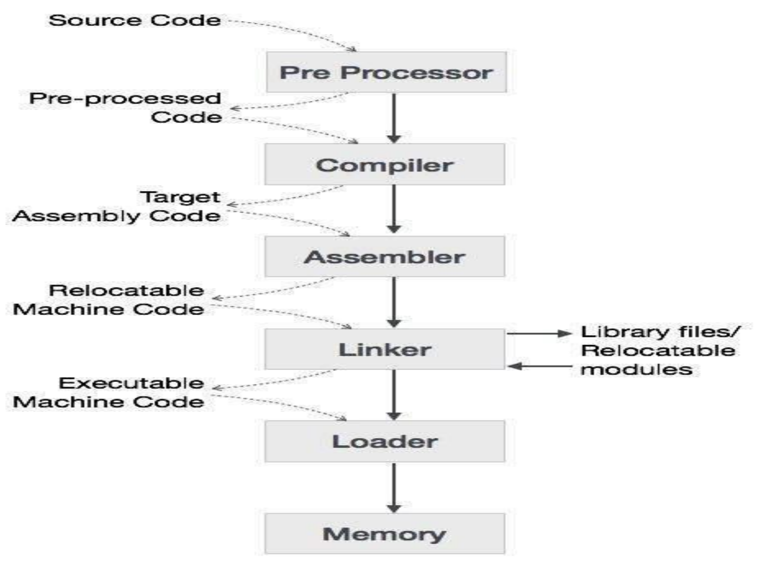

Figure : Processing of the language in various phases

Assembler:
An assembler is a program that converts the assembly language to machine-level language.

Program Compilation:
User writes a program in C language (high-level language).
The C compiler, compiles the program and translates it to assembly program (low-level language).
An assembler then translates the assembly program into machine code (object).
A linker tool is used to link all the parts of the program together for execution (executable machine code).
A loader loads all of them into memory and then the program is executed.

Preprocessor:
A preprocessor, generally considered as a part of compiler, is a tool that produces input for compilers.
It deals with macro-processing, augmentation, file inclusion, language extension, etc.

Interpreter:
An interpreter, like a compiler, translates high-level language into low-level machine language.
The difference lies in the way they read the source code or input.

Assembler:
An assembler translates assembly language programs into machine code.
The output of an assembler is called an object file, which contains a combination of machine instructions as well as the data required to place these instructions in memory.

Linker:
Linker is a computer program that links and merges various object files together in order to make an executable file.
All these files might have been compiled by separate assemblers.

Loader:
Loader is a part of operating system and is responsible for loading executable files into memory and execute them.
It calculates the size of a program (instructions and data) and creates memory space for it.
It initializes various registers to initiate execution.

Cross Compiler:
A compiler that runs on platform (A) and is capable of generating executable code for platform (B) is called a cross-compiler.

Source to Source Compiler:
A compiler that takes the source code of one programming language and translates it into the source code of another programming language is called a source-to-source compiler.

Tutorial Sheet Questions for Practice

1. Give Example of Compilers?

……

……

……

……

2. Which compiler is used by Python Programming Language? Give some details about it?

……

……

……

……

3. Why do we need compiler?

……

……

……

……

4. What are the various components of a compiler?

……

……

……

……

……

……

Student Name: ……………………………………………………………………University ID……………………………