1. What is a compiler? (1 Mark)

• Clearly define what a compiler is and explain its role in translating source code into

machine code.

In computing, a compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language). The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a low-level programming language (e.g. assembly language, object code, or machine code) to create an executable program.

A compiler takes the source code as a whole and translates it into object code all in one go. Once converted, the object code can be run at any time. This process is called *compilation*.

All of the object files used in a program must be combined before the program can be run. This is done using a linker tool, which takes one or more objects and groups them into a single *executable* or a *library*.

Compilers have several advantages:

* Compiled programs run quickly since they have already been translated.
* A compiled program can be supplied as an executable file. An executable file is a file that is ready to run. Since an executable file cannot be easily modified, programmers prefer to supply executables rather than source code.
* Compilers *optimise* code. Optimised code can run quicker and take up less *memory* space.

Compilers have several disadvantages:

* Because the source code is translated as a whole, there must be enough memory space to hold the source code, the compiler and the generated object code. There also needs to be temporary working space for the compiler to perform the translation. Modern systems either have enough memory or use *virtual memory* to hold all the data.
* Compilers do not usually spot errors - the program has to be compiled and run before errors are encountered. This makes it harder to see where the errors lie.
* The source code must be recompiled every time the programmer changes the program.
* Source code compiled on one platform will not run on another - the object code is specific to the *processor's* architecture.

How does a compiler work?

The process of compiling code involves several steps.Basically, Code passes through these steps sequentially and if there is any mistake in code then it will be examined through these steps and thus compilation process stops in-between and show compilation error.Oherwise if everything is OK then compiler does not show any error and compile the source code . There are the following steps which are involved during compiling:

Lexical Analysis

The source code is first tested by the compiler’s lexer, which breaks the source code into tokens, such as keywords, identifiers, operators, and punctuation.

Syntax Analysis

The next step is syntax analysis, where the compiler’parser checks the code for syntax errors and ensures that it follows the rules of the programming language. The compiler generates an Abstract Syntax Tree (AST) that represents the structure of the code.

Semantic Analysis

Once the code is syntactically correct, the compiler performs semantic analysis on parsed code to find the meaning. The compiler check for logical errors, such as type mismatches, undeclared variables, and incorrect usage of operators.

Optimization

The compiler may perform various optimizations to improve the performance of the resulting code. It is an optional phase of compiler. It removes dead code and arranges the sequence of instructions in order to boost the program execution.

Code generations

The last step is code generation, where the compiler translates the AST into machine-readable code. The code generator creates assembly language code, which is then translated into binary code that can be executed by the computer.

b. What is the relationship between Java and JavaCC? (2 Marks)

• Explain how JavaCC is used in Java to build parsers and why it is relevant in the

context of compiler construction. Why can we include java code in javacc source file?

c. What is a parser? (1 Mark)

• Provide a clear definition of a parser and its role in analysing and structuring input

code.

d. What is a token? (1 Mark)

• Explain the concept of a token, its significance in parsing, and how tokens are used by

the parser.