Basic Programming Terminology

Before moving forward, let's understand the various programming terminologies that we use in our day-to-day life:

1. Keywords: Keywords are predefined words in a programming language that has special meaning and cannot be used as a variable name or function name. Examples of keywords in C++ include "int", "float", "while", and "for".

2. Variables: Variables are named storage locations in a program that can hold a value of a particular data type. In C++, variables must be declared with a specific data type before they can be used. Examples of data types in C++ include "int" for integers, "float" for decimal numbers, and "char" for characters.

3. Functions: Functions are self-contained blocks of code that perform a specific task. In C++, functions must be declared before they can be used. Functions can take input parameters and return a result.

4. Object Oriented Programming (OOP): OOP is a programming paradigm that organizes code into objects that represent real-world entities. These objects can have attributes (variables) and behaviors (functions). OOP languages, such as C++, allow for the creation of classes, which are templates for objects.

5. Statically Typed Languages: Statically typed languages are those in which the type of a variable must be declared at the time of its creation and cannot be changed later. C++ is a statically typed language.

6. Dynamically Typed Languages: Dynamically typed languages are those in which the type of a variable is not declared at the time of its creation and can be changed later. Examples of dynamically typed languages include Python and Ruby.

7. Header Files: Header files in C++ are files that contain declarations for functions, variables, and other constructs that can be included in a C++ program using the preprocessor directive "#include". These declarations allow the program to use the functions and variables defined in the header file without having to know their implementation details.

8. Namespace: A namespace in C++ is a container for a set of identifiers (variables, functions, etc.) that helps to prevent naming conflicts in large programs. Namespaces are declared using the "namespace" keyword and can be referenced using the "::" operator.