**analysis -**  Thinking about a problem and possible solutions

**class -** user-defined type that contains member functions, member data and member types

**class member -** functions, data and type contained in class

**data member -** include all members that can be defined in class with any of fundamental types

**design -** Thinking about what parts a program should have and how they need to be connected

**divide by zero -** impossible mathematical operation that causes a termination of program

**grammar -** a definition for the syntax of input

**implementation -** writing a code, then debugging and testing

**interface -** shows how a piece of code (like function or a class) should be called

**member function -** are declared inside a class and can process the data within a class

**parser -** reads a stream of tokens according to the grammar

**private -** part of class with the details that doesn’t directly accessible to users

**prototype -** first version of a program

**pseudo code -** logic of program described in informal language

**public -** part of class that is directly accessible to users

**syntax analyzer -** a parser

**token -** a sequence of symbols that we consider as a unit

**use case -** use of program to demonstrate its purpose and abilities