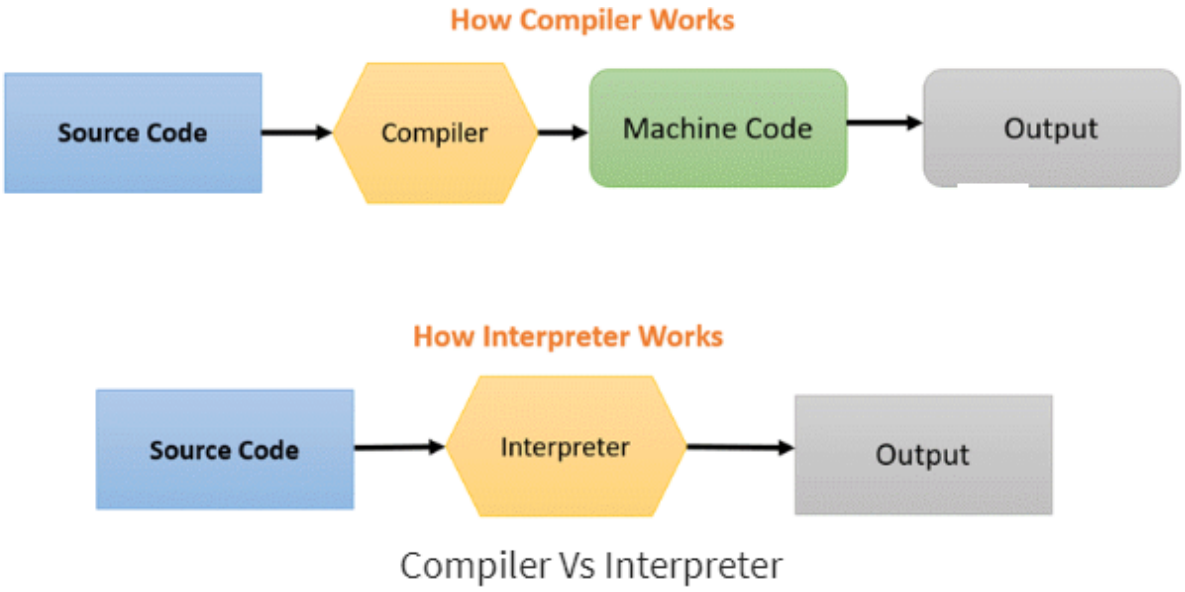
Lab 01

Introduction To Compiler

Outline

1. Difference between interpreter and compiler
2. Phases of c++ Phase and command line execution
3. Compiler Phases
5. C++ Revision with use case
6. Make

Approaches of Implementing programming Language:



Compiler Vs Interpreter

Compiler (offline)

Example language: C++, Java, ... etc

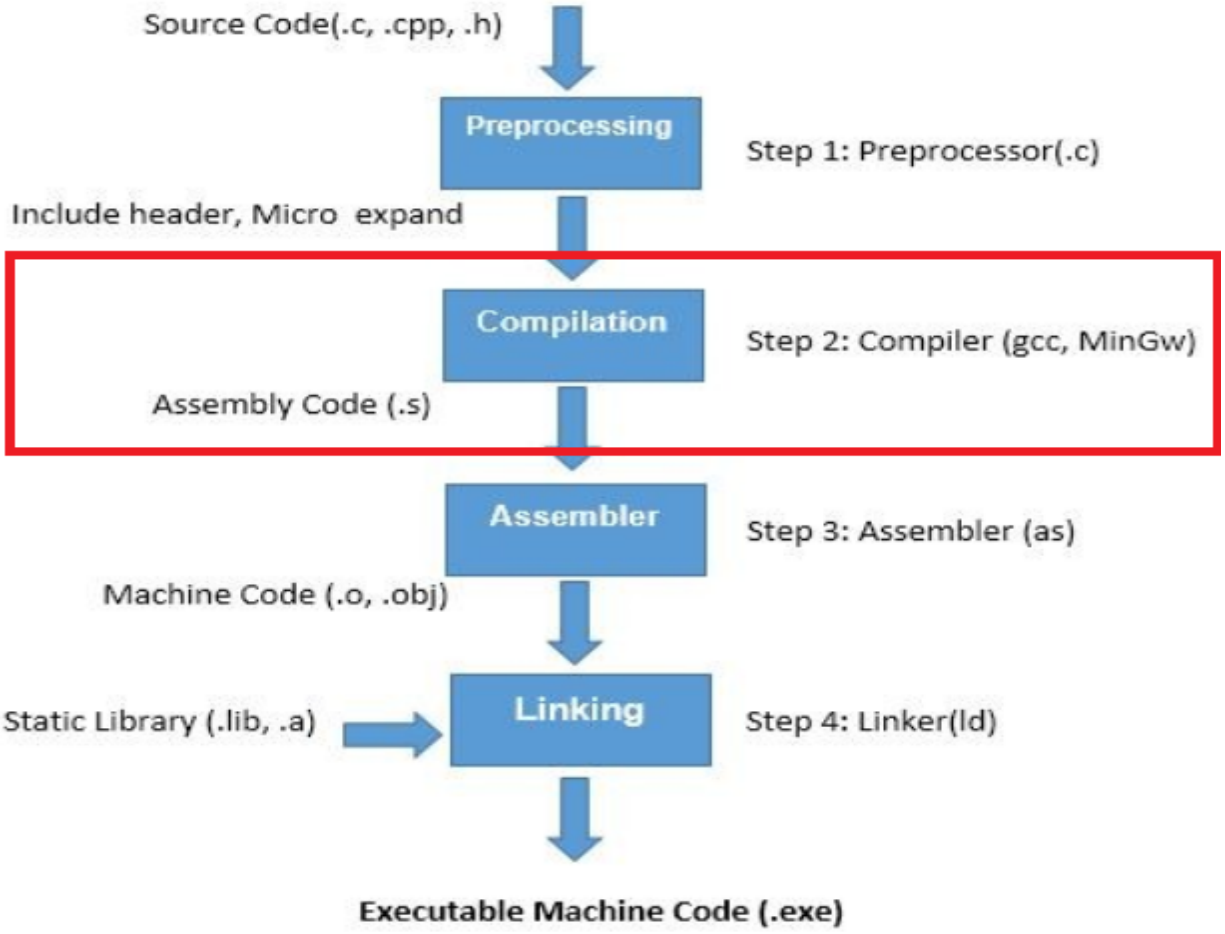
Takes as input source code and produce the machine code that takes data and produce the output

Interpreter (online)

Example language: Python,R , ... etc

Takes as input source code and data then produce the output

C++ (GCC, G++) Compilation Process



Preprocessing: ***gcc -E HelloWorld.c -o HelloWorldOutput***

(Remove comment, Expansion of marcos, Expansion of included files)

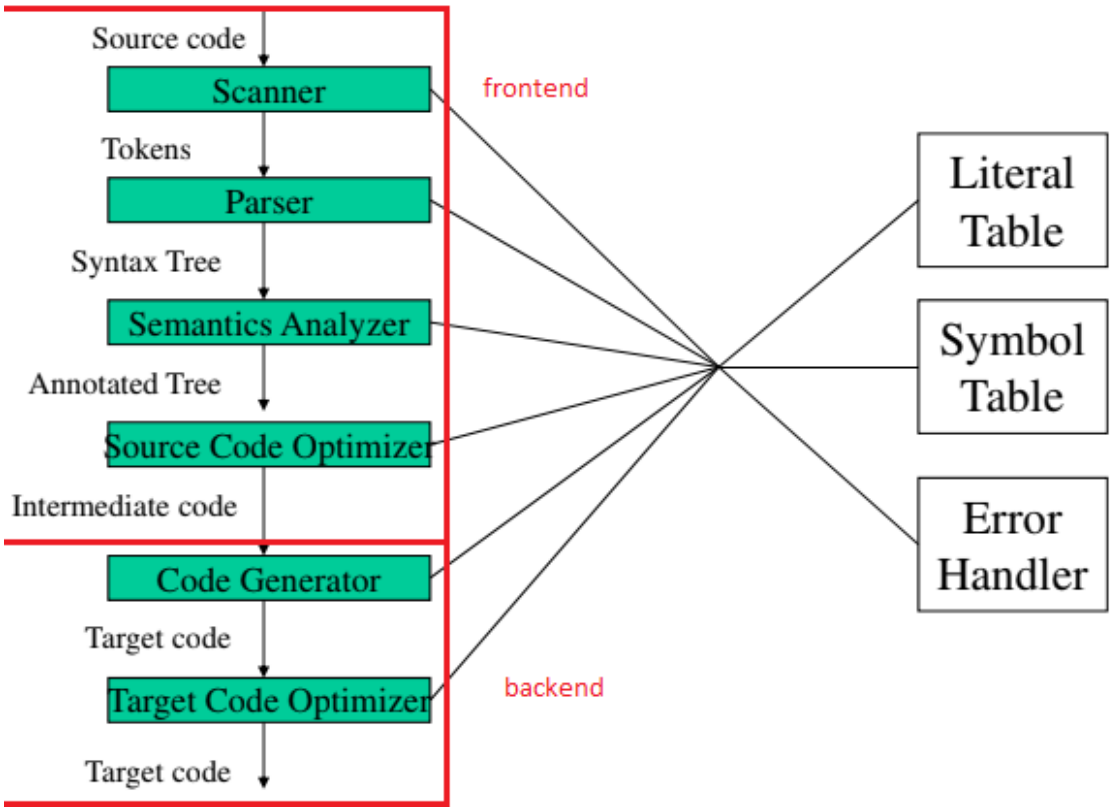
Compiler: ***gcc -s HelloWorld.c -o HelloWorld.o (code to intermediate code)***

Assembly: ***gcc -c HelloWorld.c -o HelloWorld.o*** (intermediate code to machine code)

Linking: ***gcc -o output HelloWorld.c***

(fill addresses with the actual definition)

The Phases of a Compiler



C++ Revision:

We will revise C++ and object oriented principles using use case “Creating a library Management System”

1. How to define a class in c++ and what encapsulation means?

```
class Book {
private:
    std::string title;
    std::string author;
    std::string isbn;
    bool available;

public:
    // Constructor
    Book(std::string title, std::string author, std::string isb
n)
        : title(title), author(author), isbn(isbn), available(tr
ue) {}

    // Member functions
    std::string getTitle() const { return title; }
    std::string getAuthor() const { return author; }
    std::string getISBN() const { return isbn; }
    bool isAvailable() const { return available; }
    void setAvailability(bool status) { available = status; }
};
```

2. How to Instante object?

```
Book book1("The Catcher in the Rye", "J.D. Salinger", "978031676
9174");
Book book2("To Kill a Mockingbird", "Harper Lee", "978006112008
4");
```

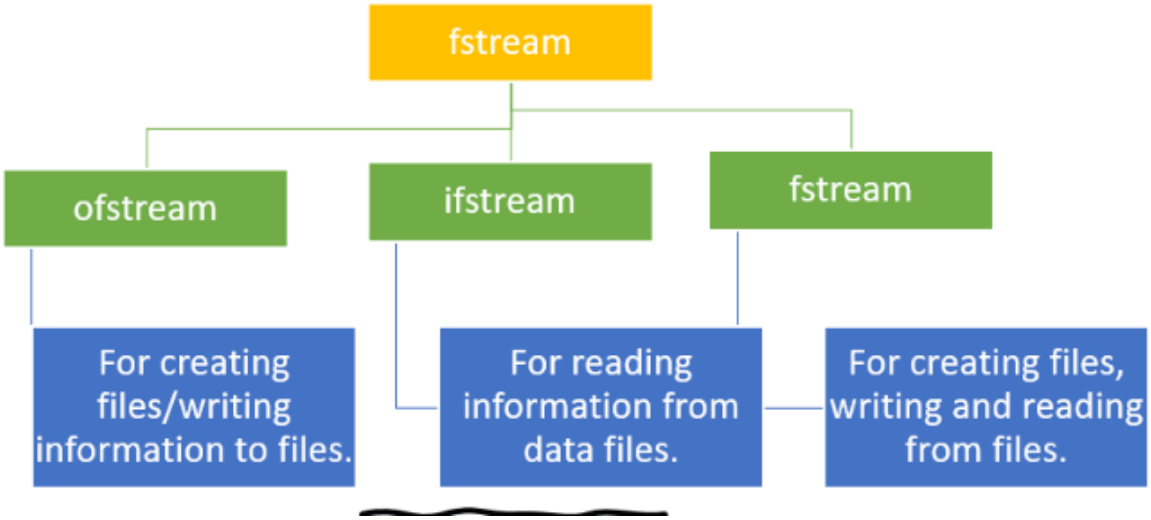
3. How to implement inheritance ?

```
class LibraryBook : public Book {
private:
    std::string borrower;
    std::string dueDate;

public:
    // Constructor
    LibraryBook(std::string title, std::string author, std::stri
ng isbn)
        : Book(title, author, isbn), borrower(""), dueDate("")
{}

    // Member functions
    std::string getBorrower() const { return borrower; }
    std::string getDueDate() const { return dueDate; }
    void borrowBook(std::string borrower, std::string dueDate) {
        setAvailability(false);
        this->borrower = borrower;
        this->dueDate = dueDate;
    }
    void returnBook() {
        setAvailability(true);
        borrower = "";
        dueDate = "";
    }
};
```

4. What is polymorphism ?

   You can apply polymorphism by creating a virtual function in the base class and override it in the derived class.

5. How to write and read from a file ?




```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
        fstream my_file;
        my_file.open("my_file.txt", ios::out);
        if (!my_file) {
                cout << "File not created!";
        }
        else {
                cout << "File created successfully!";
                my_file << "Guru99";
                my_file.close();
        }
        return 0;
}
```

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
        fstream my_file;
        my_file.open("my_file.txt", ios::in);
        if (!my_file) {
                cout << "No such file";
        }
        else {
                char ch;

                while (1) {
                        my_file >> ch;
                        if (my_file.eof())
                                break;

                        cout << ch;
                }

        }
        my_file.close();
        return 0;
}
```

```
// open a file to perform read operation using file object.
new_file.open("xyz.txt", ios::in);

// Checking whether the file is open.
if (new_file.is_open()) {
    string sa;
    // Read data from the file object and put it into a string.
    while (getline(new_file, sa)) {
        // Print the data of the string.
        cout << sa << "\n";
    }

    // Close the file object.
    new_file.close();
```

Advanced C++ :

It is always advisable to separate the header and cpp file. Back to our library management system the files will look like the following:

```h
#include<iostream>
#ifndef BOOK_H
#define BOOK_H

using namespace std;
class Book {
private:
    string title;
    string author;
    string isbn;
    bool available;
public:
    Book(string title, string author, string isbn);
    string getTitle() const;
    string getAuthor() const;
    string getISBN() const;
    bool isAvailable() const;
    void setAvailable(bool status);
    void print();
};

#endif
```

```cpp
#include "Book.h"
#include <iostream>
Book::Book(string title, string author, string isbn): title(title), author(author),
isbn(isbn), available(true){}
string Book::getTitle() const { return title; }
string Book::getAuthor() const { return author; }
string Book::getISBN() const { return isbn; }
bool Book::isAvailable() const { return available; }
void Book::setAvailable(bool status) { available = status; }
void Book::print() {
    cout<<"The book with title: " << title << " ,author: "<< author << " ,isbn: "<<isbn<< " is ";
    if(available) cout<< " available "<<endl;
    else cout<<"not available"<<endl;
}
```

```h
#ifndef LIBRARYBOOK_H
#define LIBRARYBOOK_H
class LibraryBook: public Book{
private:
    string borrower;
    string dueDate;
public:
    LibraryBook(string title, string author, string isbn);
    string getBorrower() const;
    string getDueDate() const ;
    void borrowBook(string borrower, string dueDate);
    void returnBook();
};
#endif
```

```cpp
#include "LibraryBook.h"
#include "Book.h"
#include <iostream>

LibraryBook::LibraryBook(string title, string author, string isbn): Book(title, author, isbn),
borrower(""), dueDate(""){}
string LibraryBook::getBorrower() const { return borrower; }
string LibraryBook::getDueDate() const { return dueDate; }
void LibraryBook::borrowBook(string borrower, string dueDate) {
    setAvailable(false);
    this-> borrower =  borrower;
    this-> dueDate = dueDate;
}

void LibraryBook::returnBook() {
    setAvailable(true);
    borrower = "";
    dueDate = "";
}
```

```cpp
#include "Book.h"
#include "LibraryBook.h"
using namespace std;

int main()
{
    Book b1("The catcher in the Rye", "J.D.Salinger", "978031676917");
    LibraryBook libraryBook("1984", "GeorgeOrwell", "9780451524935");
    libraryBook.borrowBook("John Doe", "2023 - 10- 31");
    libraryBook.print();
    cout<<"Borrower: "<< libraryBook.getBorrower() <<endl;
    cout<<"Due Date: " << libraryBook.getDueDate() <<endl;
    libraryBook.returnBook();
    return 0;
}
```

In order to compile these file using g++ you should mention all cpp files like the following command:

``` ***g++ Book.cpp LibraryBook.cpp main.cpp*** ```

This is still good but as our project is getting bigger this command is getting bigger and the dependencies are getting complicated then ```make``` and ```cmake``` solved this problem.

What is makefile ?

It is automated build process for c++ application

Example on cMake:

In the main directory of project create file with name makefile (note in order to execute the file you must download make to your system first)

```
libraryManagementSystem: Book.o LibraryBook.o main.o
    g++ Book.o LibraryBook.o main.o -o libraryManagementSystem

Book.o: Book.cpp
    g++ -c Book.cpp

LibraryBook.o: LibraryBook.cpp
    g++ -c LibraryBook.cpp

main.o: main.cpp
    g++ -c main.cpp
```

To run
``` ***make*** ```

Materials & Resources

- Compiler vs Interpreter – Difference Between Them (guru99.com)
- GCC compiler how it works? | LinkedIn
- C Build Process in details | LinkedIn
- Everything you want to know about GCC | by megha mohan | Medium
- C++ File Handling: How to Open, Write, Read, Close Files in C++ (guru99.com)
- Makefile Tutorial By Example