# LAB # 06

**Name : Adeesha Sherani**

**Sap Id : 47379**

**Submitted to : Miss Kousar**

**Subject: Operating System**

# Task

**Q1:** Explain the process of compiling a C program in Linux. What command is used to compile the program?

- Sudo apt install.
- Sudo apt install gcc.
- Sudo apt install g++.
- Sudo apt update.
- Create a file for example nano file.cpp
- GNU screen appears write C program.
- Press ctrl X.
- Press Y for yes. And then press Enter.
- Write chmod 777 file.cp to give all commands to file.
- Write gcc file.cp -o lab (this command compiles the C file file.cp and create an executable file name lab).
- Write ./lab to see output.

**Q2:** What is the purpose of the `-o` option in the `gcc` command? Provide an example.

The -o option in the gcc command allows us to specify the name of the output file. Without this option the default name of the output file has been created like **a.out**. With this option we can give the name to the output file itself.
For example,

```
gcc -o my_program my_program.c
```

if we execute gcc **file.c -o myprogram**, it will compile the file.c file and creates an executable name myprogram instead of the default **a.out**.

**Q3:** What is the difference between `g++` and `gcc`? When would you use each?

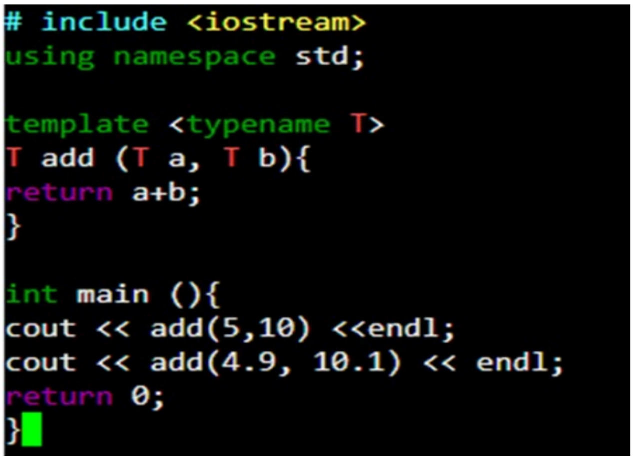
The difference between g++ and gcc is that gcc is used for compiling C programs while g++ is specially used for compiling C++ programs. Although gcc can compile C++ program it doesn't link directly with C++ libraries. We use g++ for C++ program compilation and gcc for C program compilation.

**Q4:** How do you compile and run a C++ program from the terminal? Provide the necessary commands.

- Chmod 777 file.cpp
- g++ file.cpp -o lab
- ./lab

**Q5:** What are templates in C++ in Linux? Write a simple example of a function template.

Templates in C++ can create functions or classes that can work with any types of datatypes. So, we don't need to rewrite the same code for different types like int, float, etc. In simple, we can say it makes our code flexible.



```
# include <iostream>
using namespace std;

template <typename T>
T add (T a, T b){
return a+b;
}

int main (){
cout << add(5,10) <<endl;
cout << add(4.9, 10.1) << endl;
return 0;
}
```

**Q6:** Discuss the significance of file extensions in C programming. Why should source files be saved with `.c` or `.cpp` extensions?

In C programming file extensions like .c for C and .cpp for C++ are necessary because they tell the compiler what kind of code we're working with.

**Q7:** What are the common errors that can occur when compiling C programs, and how can they be resolved?

**Syntax Error:**

Missing semicolons, unmatched brackets.

**Solution:**

Recheck the code properly and correct it.

**Missing Libraries:**

Like if I'm writing cin and cout but not including iostream library

**Solution:**

Make sure we include all the required libraries.

**Linker Error:**

Function is declared but not defined.

**Solution:**

Make sure all functions are defined.

**Q8:** Explain how you can manage permissions for an executable file in Linux. What command is used for this purpose?

In Linux we can manage permissions for an executable file by using chmod command who can read, write and execute.

**Q9:** What is tarball, and what advantages does it offer for distributing software on Linux? Discuss the limitations of using tarballs for software installation and management.

A tarball is an archive file created by used tar command. It is the first file introduced for Linux. We used an untar command to extract file. The advantages of using tarballs include reducing the size of files for faster downloads. It has multiple files and directories in one file making it distributive. Some limitations of tarballs don't tell the version of software if software had other libraries or tools we had to find, install and update them by yourself, doesn't tell file location and dependencies.

**Q10:** Explain the purpose of the RPM package format and how it addresses the shortcomings of tarballs.

When we install an RPM (red hot package manager) package, it automatically resolves and installs required libraries or tools. RPM also allows easy uninstallation of software, addressing the shortcomings of tarballs, which require manual handling. RPM gives additional information like a version of software, file list of file packages and they will or where installed and tarballs doesn't provide this information.