**CONTENT ASSESSMENT 8**

**Name:** Naseer Almargan

**Date:** 06-07-2020

**1. Explain how to use makefiles and CMake to build libraries of code for scalable C/C++ projects.**

**Ans:**

**INTRODUCTION TO USING CMake FOR BUILDING C++ PROJECTS:**

The make utility and Makefiles provide a build system that can be used to manage the compilation of programs irrespective of the programming language they are written in. However, it can become complex when projects are deployed on varying platforms.

Here CMake comes in picture. It is a cross-platform Makefile generator. It can automatically generate the Makefiles for your project.

**Example:**

Here is the sample helloworld.cpp Code

#include<iostream>

using namespace std;

int main (int argc, char \*argv[]) {

cout << "Hello Everyone!";

return 0;

}

And here is the CMakeLists.txt File Code: -

cmake\_minimum\_required (VERSION 2.8.9)

project (hello)

add\_executable (hello helloworld.cpp)

**EXPLANATION OF THE CMakeLists.txt FILE: -**

* First-line sets the minimum version of CMake we are using for this project.
* Second-line sets the name of the project.
* Third-line is for requesting that an executable be built using the mentioned source file. The name of executable here is hello while the second argument is source i.e. helloworld.cpp.

**Note: -**

For the project to work, you need to make sure that CMake is installed.

Now just execute the **cmake** command and pass it the directory which contains your source code and the CMake file.

Once the Makefile is created, use the **make** command to build the project.