LOVELY PROFESSIONAL UNIVERSITY

Transforming Education Transforming India

SIX WEEKS SUMMER TRAINING REPORT

on

C++ for C Programmers Part-A

Submitted by

Sai Kiran**.T**

Registration No : 11608063

Programme Name : B-Tech CSE(2016-2020)

School of Computer Science & Engineering Lovely Professional University, Phagwara

(June-July, 2018)

DECLARATION

I hereby declare that I have completed my six weeks summer training in coursera(which is an online portal) from 6th June 2018 to 7th July 2018 under the guidance of Professor Iran Pohl. I have declare that I have worked with full dedication during these six weeks of training and my learning outcomes fulfill the requirements of training for the award of degree of C++, Lovely Professional University, Phagwara.

(Signature)

SaiKiran.T

11608063

Acknowledgement

I would like to express my very great appreciation to Ira pohl for his valuable and constructive way of teaching throughout the planning and development of this technology. His willingness to give his time so generously to teach in very easy way has very much appreciated.

I would also like to thank the staff of the following organization for enabling me to complete my course and for helping me with every queries:

coursera organization

Table of Content

1.Introduction

This is the project report for the summer training, Here project report is done for c++ technology which I have learnt in this summer, I learnt this c++ technology from an organization called coursera, This course is of 5 weeks where course name is c++ for c programmers,

In this course I had learnt about the following topics.

Topics

How to convert an existing C program to C++

Use of type safe input/output

Dijkstra's shortest path algorithm

C++ functions and generics

C++ classes and OO

Point: Default constructor and initializing syntax

Conversion constructors

Copy constructor

List and dynamic memory allocation

Prim's and Kruskal's algorithms

Use of basic container classes

Tripod-container, iterator algorithm

These are the topics which I have learnt in summer training.

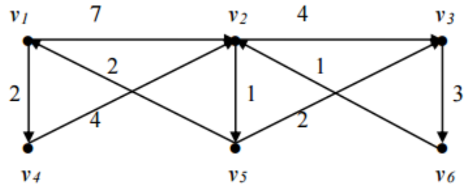
Dijkstra's Algorithm

Dijkstra's algorithm has many variants but the most common one is to find the shortest paths from the source vertex to all other vertices in the graph.

Algorithm Steps:

- Set all vertices distances = infinity except for the source vertex, set the source distance = 0.
- Push the source vertex in a min-priority queue in the form (distance , vertex), as the comparison in the min-priority queue will be according to vertices distances.
- Pop the vertex with the minimum distance from the priority queue (at first the popped vertex = source).
- Update the distances of the connected vertices to the popped vertex in case of "current vertex distance + edge weight < next vertex distance", then push the vertex with the new distance to the priority queue.
- If the popped vertex is visited before, just continue without using it.
- Apply the same algorithm again until the priority queue is empty.

Problem 10: Given the graph below, use Dijkstra's algorithm to find the shortest path from the top-left corner vertex to the bottom-right corner vertex. Label the vertices accordingly. Give the length of the shortest path.

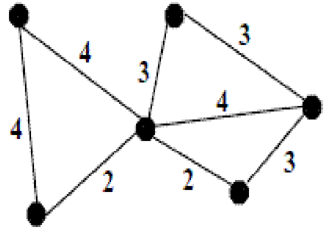
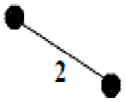
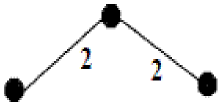
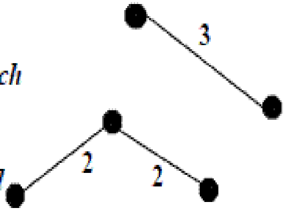
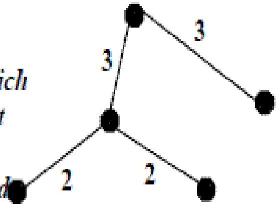
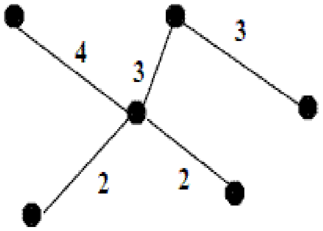


S: {v1,v4,v2,v5,v3,v6}

2 6 10 2 7 12

S	L					
	v1	v_2	v_3	v_4	v_5	v_6
{ v_1 }	**2**	∞	∞	∞	∞	∞
{v2}	**2**	**7**	∞	**2**	∞	∞
{v3}	**2**	**6**	∞	**2**	∞	∞
{v4}	**2**	**6**	**10**	**2**	**7**	∞
{v5}	**9**	**6**	**10**	**2**	**7**	∞
{v6}	**9**	**6**	**10**	**2**	**7**	**12**

Kruskal's Algorithm

1 *Given a network…………*  	2 *Choose the shortest edge (if there is more than one, choose any of the shortest)……..*  	3 *Choose the next shortest edge and add it……..* 
4 *Choose the next shortest edge which wouldn't create a cycle and add it.*  	5 *Choose the next shortest edge which wouldn't create a cycle and add it.*  	6 *Repeat until you have a minimal spanning tree.* 

2.Technology learnt

Technology which I learnt is C++

C++ is a multi-paradigm programming language that supports object oriented programming (OOP) created by Bjarne Stroustrup in 1983 at Bell labs, C++ is an extension of C programming and the programs written in C language can run in C++ compiler. The development of C++ actually started four years before its release, in 1979. It did not start with the name C++. Its first name was C with classes.In the late part of 1983, C with classes was first used for AT&T's internal programming needs. Its name was changed to C++ later in the same year. It is of course also used in a wide range of other application domains, notable graphics programming. C++ supports inheritance through class derivation. Dynamic binding is provided by Virtual class function.

USES OF C++ LANGUAGE

C++ is used by programmers to develop computer software

It is used to create general system software

Used to build drivers for various computer devices

Software for servers and software for specific applications

Used in the creation of video games.

It offers performance and speed.

This is the power of C++.

It plays well with others. C++ can interface with nearly any other language.

And almost any system can compile and run C++ code.

It’s “mid-level” because it’s got a combination of high- and low-level features.

It’s one of the programming languages the MongoDB database is written in.

ADVANTAGE OF C++

C++ is relatively-low level and is a systems programming language.

It has a large community.

It has a relatively clear and mature standard.

Modularity

Reusability and readability

Applications of C / C++ in the Real World

A super set of C, C++ is an object-oriented programming language and incorporates all the features offered by C. C++ started its journey as C with classes. Gradually, it has evolved and despite the popularity of other programming languages like C# and Java, C, C++ holds its own as one of the most widely used languages for scripting. In applications,

C++ is ubiquitous in Real World c++ applications

1. Games:
2. Graphical user interface (G.U.I) applications
3. Web browsers
4. Advance computations and graphics
5. Database software
6. Operating systems
7. Enterprise software
8. Compilers
9. Medical and engineering applications

Scope of c++ in real world

C++ is one of the most reliable programming language, where the scope of c++ in real world very high.

All the major operating systems (Linux (actually C but we can be lenient), Mac OS X, Windows)

All the Blizzard games (World of Warcraft, Diablo series, StarCraft series...)

Nearly all the other AAA games too (Unreal Engine & Unity are both C++ at core. Valve and CryTek i.e. CryEngine are also C++)

Most console games be it for Nintendos or for Playstation.

Most interpreted programming languages have VMs written in C or C++ (Java, Python, Ruby, Perl, PHP, Lua)

Many development environments and compilers (GNU C++, Visual Studio, KDevelop, but not Eclipse)

Most productivity software (MS Office, Photoshop, Maya/3DS, SolidWorks, Inventor, AutoCAD)

Major web browsers (Chrome, IE, Firefox, Opera... Safari might use substantial amounts of Objective C, though)

C is the foundational language of modern computing.

C++ is itself usually written in C. When you make a new operating system or new computing hardware,your first task is usually to write a simple C compiler. Once you have that, you use it to create/compile a powerful C compiler, and you use that to build a C++ compiler. From there, you can do anything--the other languages are built on top of this foundation.

C and its successor C++ are leveraged for diverse software and platform development requirements, from operating systems to graphic designing applications. Further, these languages have assisted in the development of new languages for special purposes like C#, Java, PHP, Verilog etc. As updating of these languages, particularly C++, continues on a periodic basis, their utilization for robust applications is likely to expand as well.

If you are crazy/passionate/bored enough to create a 3d game engine from scratch, then this language is perfect. But you can easily use the already made game engines to help you produce a game. You don't even need to know how to code with unreal. Coding in the unreal game engine 4 is an option, but u really do not need it.

C++ helps and these are the scopes of c++.

Reason for choosing c++ technology

Especially in my case I am not that good coder so to make my basics very strong and to start real world implementation I choosed c++ as a summer training.

C++ is one language among many, but its unusual combination of strengths make it an essential tool for the serious engineer. Although you may take years to fully understand some of its features, you can benefit from its power immediately.

So as c++ helps even intermediates to develop I preferred c++.

And in c++ there are few unique features to grab that such as

1. Compared to many of the other OO languages (i.e. Java, C# etc.), C++ has better support for the functional programming style.
2. Good support for generic programming.
3. C++ is faster. No matter what you have heard, the overall speed of C++ applications (either real or perceived) is greater than that of other languages.
4. Templates and generic programming without too much brain twisting. Templates and type safety is very important. If C++ hasn't had templates, it would not be used.
5. Direct interfacing with C and O/S routines.
6. Plethora of good libraries.
7. C++ is always the right choice for serious applications that can be re-compiled on multiple platforms.
8. C++ is preferred choice of programmers all over the world when it comes to programming for creating games.

Learning outcome from Training /Technology learnt

How this technology helps you in future?

For me this course is to improve basics and fundamentals about c++ programming especially, this technology consists of

How to convert an existing C program to C++

Use of type safe input/output

Dijkstra's shortest path algorithm

C++ functions and generics

C++ classes and OO

Point: Default constructor and initializing syntax

Conversion constructors

Copy constructor

List and dynamic memory allocation

Prim's and Kruskal's algorithms

Use of basic container classes

Tripod-container, iterator algorithm

These are the topics which I have learnt in summer training.

While this course I wrote 3 pre-graded assignments so these are the following

1: Convert a C program to C++

2: Implement Dijkstra's Algorithm

3: Compute the minimum spanning tree for an inputted graph

So every assignment which I have done is linked to real time applications.

Here while doing first week they thought me from how this c++ works few of their functionalities and mainly while coming to the end of week they thought and they made me prepared for converting c program to c++

While doing second week they started taking this course to the depth started with generics and functions ,template arguments, here they introduced me to the graph theories ,Algorithms, operator overloading and so on they introduced these all in a overview

While doing third week they started with c++ overloading and object orientation ,They thought the usage of Dijkstra’s algorithm very briefly at the end of second week they prepared me to implement Dijkstra's Algorithm.

While doing fourth week they thought me about prim’s and kruskal’s algorithm’s by the end of this week they made me ready for computing the minimum spanning tree for an inputted graph.

Learning objectives and things which I learnt in this course

1.I came to know how an existing c++ program works

2.I learnt to discover errors in a c++ program and came to know how to fix them

3.I learnt to analyse a problem and construct a c++ program that solves it.

How this technology will help me in future

As my future projects are based on application development and database, software development this technology is going to help in thoroughly.

As c++ concepts are base for many other languages I can easily learn other courses which I want to.

As this is technology is ever green I can also do many projects based on c++.

May I can also work as a c++ developer.

GANTT CHART

week no.	Learning/Assignment	Quiz	Duration	start	finish
1.	yes	nill	7 days	06/06/2018	12/06/2018
2.	yes	nill	7 days	13/06/2018	20/06/2018
3.	yes	nill	7 days	21/06/2018	27/06/2018
4.	yes	nill	7 days	28/06/2018	04/07/2018
5.	nill	yes	2 days	05/07/2018	07/07/2018

Bibliography

1.Cousera online platform

2.Invensis online platform

3.mcla.instructure.com online platform

4.Techrepublic.com online platform

5.github online platform