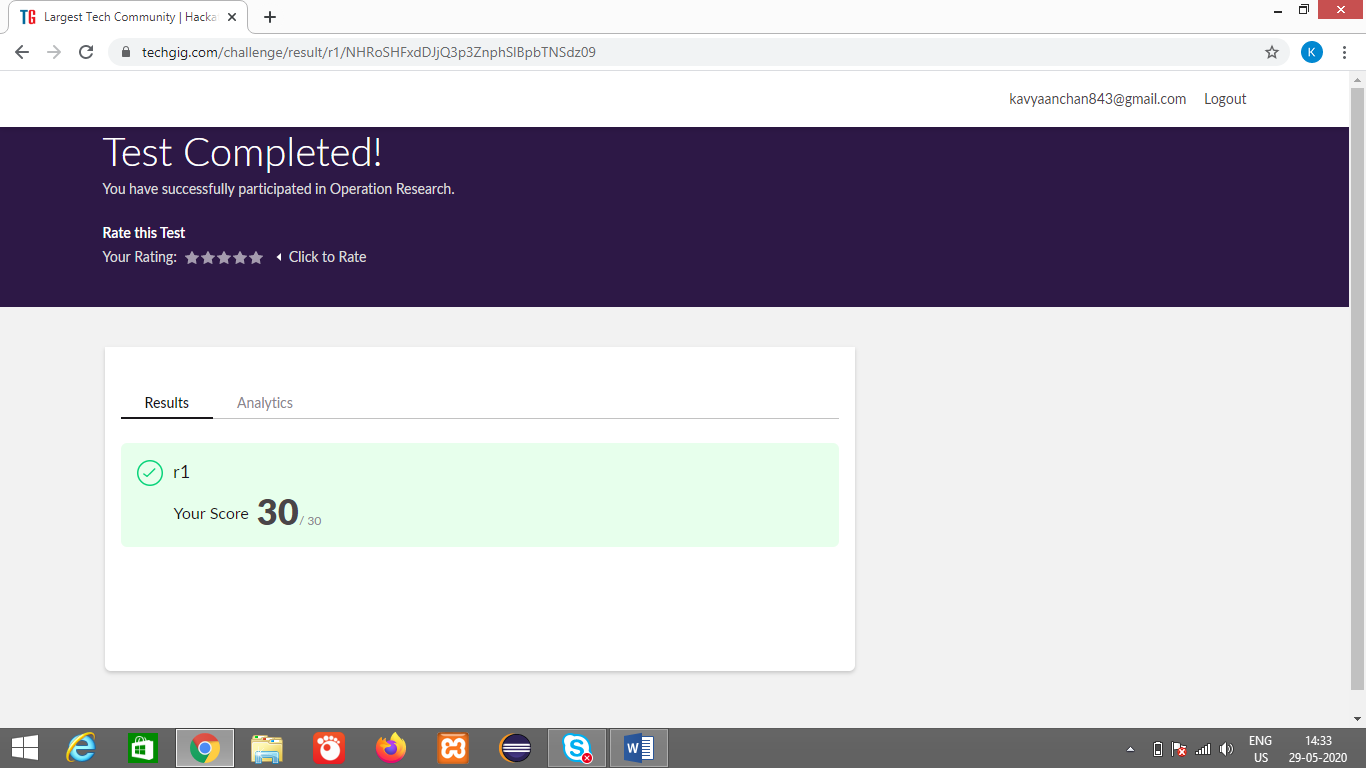
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **29-05-2020** | | | | | **Name:** | **Kavya** | |
| **Sem & Sec** | **6th & A** | | | | | **USN:** | **4AL17CS041** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **Operation Research** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **30** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Robust Qt & C++ Gui Programming 2D Graphics App Tutorial** | | | | | | | |
| **Certificate Provider** | | | **Udemy** | | **Duration** | | | **3 hours** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:1.** Python program to calculate the number of lowercase and uppercase letters in a string.  2. We are given 3 strings: str1, str2, and str3. Str3 is said to be a shuffle of str1 and str2 if it can be formed by interleaving the characters of str1 and str2 in a way that maintains the left to right ordering of the characters from each string. For example, given str1="abc" and str2="def", str3="dabecf" is a valid shuffle since it preserves the character ordering of the two strings. So, given these 3 strings write a function that detects whether str3 is a valid shuffle of str1 and str2. | | | | | | | | |
| **Status: yes, uploaded all the programs** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | [**https://github.com/kavya-077/DAILY-STATUS**](https://github.com/kavya-077/DAILY-STATUS) | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

Online Test Details: (Attach the snapshot and briefly write the report for the same)



Certification Course Details: (Attach the snapshot and briefly write the report for the same)

**This course will help us upgrade our C++ software development skills today!**

We will learn how to build an application using Qt and C++, from the ground up.

\*\***This course is now FREE**as a token of gratitude to my Udemy followers; Enjoy!\*\*

As I've put my heart and soul into producing this course, I understand it's no longer up to date with the Qt platform.

But it does still have valuable insights into robust app design, so I believe you'll find quite a bunch of useful stuff in here

C++ is a very powerful and versatile programming language.

Qt is multi-platform framework, the perfect choice for open source GUI development with C++.

Before investing in this course, please watch the **FREE**lectures (30 minutes) to:

* see the final app in action (check out the Final App DEMO lesson)
* decide if this course is really the right one for you

With this course, you will design and create your own C++ program for the Windows platform.

We will add new features to the app, step by step.

We will learn how to write **good source**code, at professional-level.

We can learn programming in C++ from anyone you wish to, there are many teachers and a lot of free know-how out there.

But if we want to learn to code like a professional, here's my advice to you:

* Choose to learn from someone who has already done professional level coding
* Choose a teacher who has a proven track record working experience with C++ and Qt
* Choose a mentor who has done this for years, and knows what it takes to create industry-level software

Here's how the course is structured. Each lecture prepares the groundwork for the next one. You will:

* write code in C++ (standard '03)
* use the Qt toolkit's UI elements to create the GUI (Graphical User Interface)
* set up the main application window
* drag and drop the required user interface elements (buttons, widgets, labels, spinners...) on the main window
* connect UI elements to UI events (mouse clicks, value changed)
* write custom event handling code that updates the painted graphic

We need to have a good understanding of OOP (Object Oriented Programming) fundamentals, as you will be adding and working with:

* class member variables
* class member functions
* setters / getters
* private / public / protected access modifiers
* encapsulation to protect the data
* constructors and destructors
* overriding virtual base class functions for Qt objects
* pointers to objects

We will **organize the C++ program logic**into header (.h) and implementation (.cpp) source files.

We will **design a simple GUI**(Graphical User Interface) with QtCreator (part of the Qt toolkit) and use the signal and slots mechanism to make the buttons (and other UI elements) respond to mouse clicks or value changed events.

We will setup a **custom rendering area**widget that integrates flawlessly with the Qt5 framework, by inheriting from the QWidget base class and overriding abstract class methods.

We wil learn how to **fix compiler errors**that can occur during development.

We will also **learn to fix linker errors**caused by missing function implementations.

We will use a bottom-up approach to **develop robust software**.

Notice the key word here - robust.

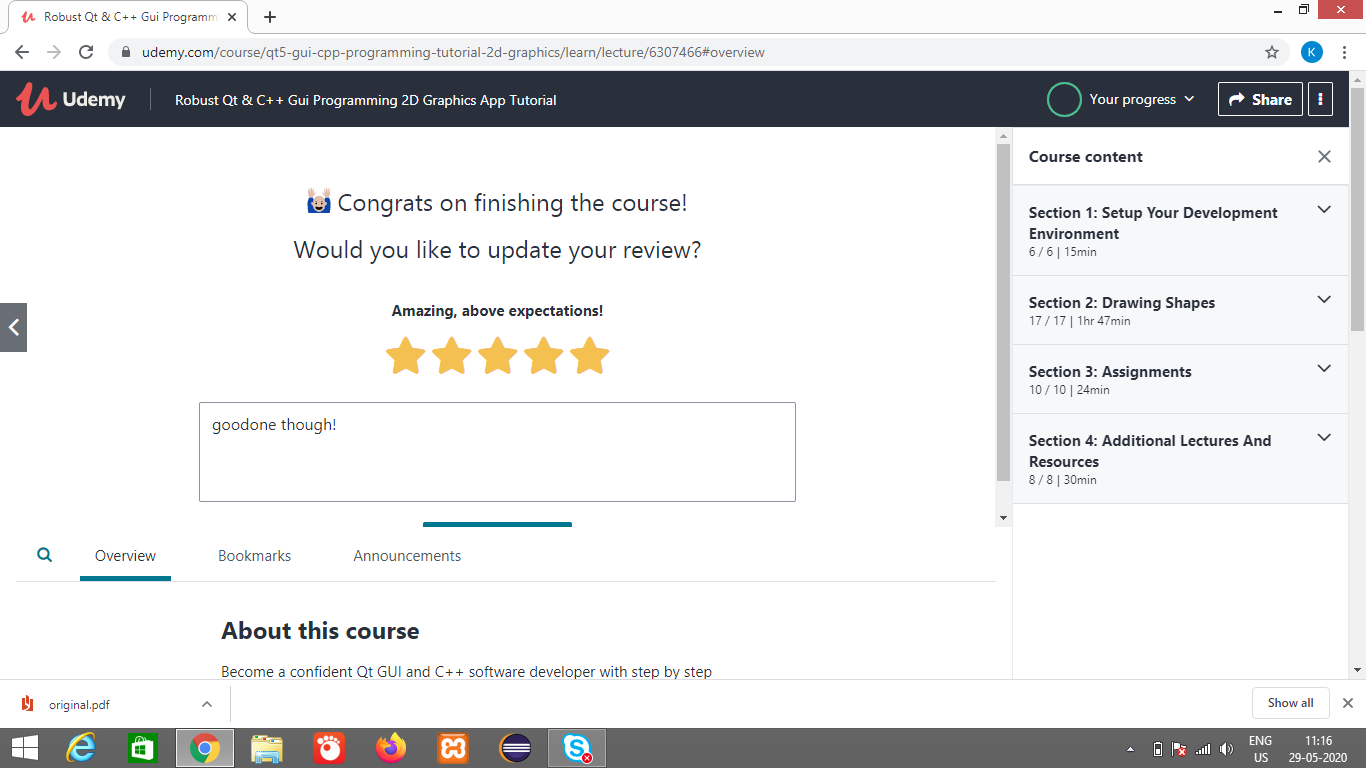
That means professional-level C++ code.

We will learn to design reusable programs with programs will be designed with reusability in mind. And will be written with future upgrade capabilities in mind.

We will also use Math and Geometry in an unexpected way, to **create unusual 2D graphics shapes**.

We will also get some hands-on experience by taking the **course assignments**.

The assignments are designed to get you comfortable with the code and teach us to **extend the application**on your own, after we have completed the course.



<https://github.com/kavya-077/DAILY-STATUS>

<https://github.com/alvas-education-foundation/kavya-onlinecourse>

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

<https://github.com/kavya-077/DAILY-STATUS>

<https://github.com/alvas-education-foundation/kavya-OCA>