Sabanci University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Summer 2017-2018

Homework 6 (BONUS) – It's Never Too Late
Due: 14 August 2018 11.55pm (Sharp Deadline)

DISCLAIMER:

Your program should be a robust one such that you have to consider all relevant user mistakes and extreme cases; you are expected to take actions accordingly!

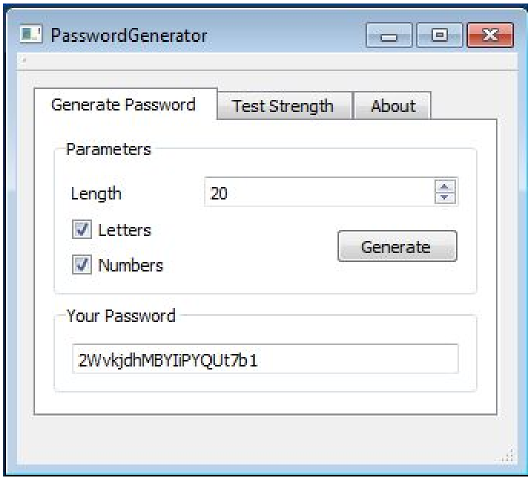
Only checking the sample run cases might not be sufficient as your solution will be checked against a variety of samples different than the provided samples; however checking these cases are highly encouraged and recommended.

You can NOT collaborate with your friends and discuss your solutions with each other. You have to write down the code on your own. Plagiarism will not be tolerated AND cooperation is not an excuse!

Introduction

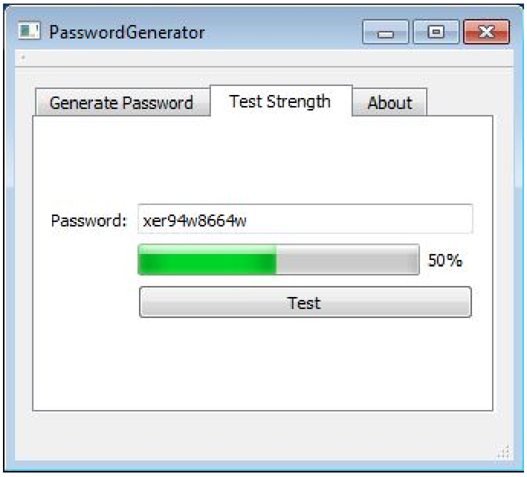
In this homework you're expected to develop a graphical user interface (GUI) based application in C++ using Qt Framework and QtCreator IDE.

You're expected to develop a simple password generator and password strength testing application as shown in the following snapshots.



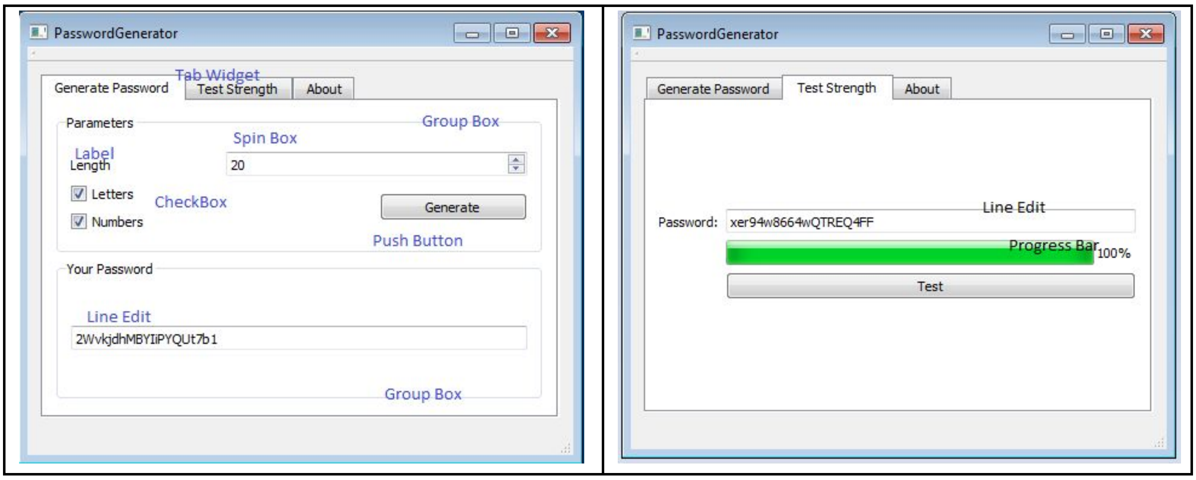
Length 20
☑ Letters
☑ Numbers

2WvkjdhMBYIiPYQUt7b1



Password: xer94w8664w
50%

The application is comprised of various common UI interface components referred to as widgets, e.g. Push Button Widget, Tab Widget, Progress Bar Widget, Line Edit, Spin Box etc. All of the widgets used have been marked for your information and can be seen in the following snapshots. All these widgets readily available for usage in the design view of Qt creator through a dra g and drop interface.



Length 20

☑ Letters

☑ Numbers

2WvkjdhMBYIiPYQUt7b1

Password: xer94w8664wQTREQ4FF

100%

The main window of the application is comprised of a Tab Widget, showing 3 tab panes (Generate Password, Test Strength, About). Clicking on any of the tab buttons should open up that pane and display the user interface of that pane.

The first pane '**Generate Password'** is responsible for generating random passwords for the user. The user can specify 3 parameters through checkboxes and a spin box widget. The spin box widget will allow the user to choose any value between *4* and *40* characters as password length.

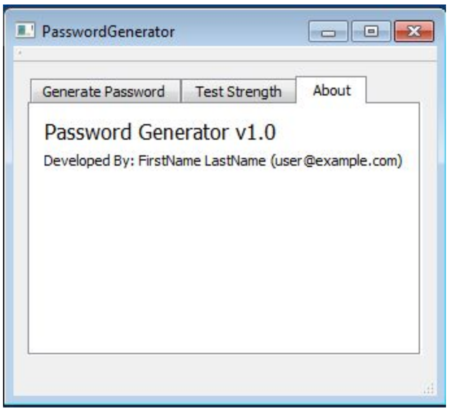
The password can be comprised solely of upper and lower case English alphabets and/or numbers that can selected by checking or unchecking the *Numbers* checkbox. However, if none of the checkboxes is selected, then upon clicking generate, the Line Edit field should remain empty.

When the user presses 'Generate' Button, (s)he should get a random password each time in the Line Edit field below on that pane.

The user interface has been arranged into Group Box containers titled 'Parameters' and 'Your password' for display organization of widgets which are related for the sub tasks. Moreover, the pane is also employing Grid Layouts to handle window resizing.

The next tab titled '**Test Strength**' is comprised of Line Edit, Progress Bar and Push button. The purpose of this pane is to determine the strength of a given password in the Line Edit into one of 4 categories (weak, normal, good and strong password) which will be displayed by 4 different progress bar levels of *25%, 50%, 75% and 100%*, respectively. For the sake of simplicity, the password strength logic is trivial, which determines the strength on the basis of the length of the password characters: Any password between *0-5* characters is *weak*, *5-10* characters is *good*, *10-15* characters is *normal*, and *15* and above will be *strong*.

Given a user password in the text field, upon pressing 'Test' button should display the strength of password as a progress bar level. Please refer to the figures above.
The next tab panel titled '**About**' is the most simplest one and its purpose is to show the software information: title, version number and developer information. You can provide your information as formatted text labels as displayed below:

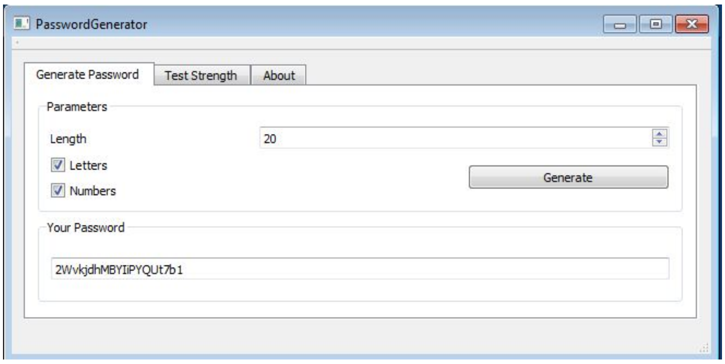


Password Generator v1.0
Developed By: FirstName LastName (user@example.com)

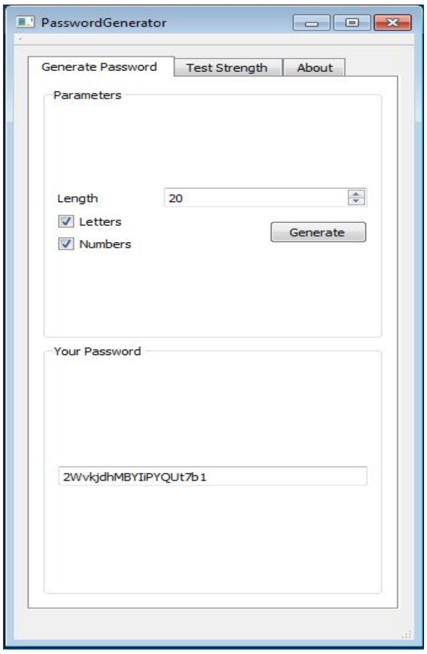
Usage of Layout Containers

Please note that automatic alignment and resizing of UI widgets is a typical issue when the main window is squeezed or stretched, so proper use of layout containers is strongly advised and it's quite convenient to use through Qt designer. You can insert any type of layout container on the form which fits your design.

The simplest layout containers that you can use are Grid Layouts, Form Layouts, Vertical Layouts and Horizontal Layouts. Your application <u>should handle resizing properly</u>. Below are two snapshots depicting two resize scenarios; without layout management, your widgets will either be cropped or stay in their original size in conjunction to large empty spaces in the main window.



Length 20
☑ Letters
☑ Numbers
2WvkjdhMBYIPYQUt7b1



Length 20
☑ Letters
☑ Numbers
2WvkjdhMBYIPYQUt7b1

Appendix: Some code snippets that maybe helpful!

Any widget on your form can be accessed by default through UI pointer. This pointer can be used to get a reference to a widget present on your form e.g.

```
//gets a reference for a checkbox with object name studentCheckBox1
QCheckBox *cb=ui->studentCheckBox1;
```

Getting a value from checkbox or setting a bool value through code:

```
bool avalue=false;
ui->myCheckBox1->setChecked(avalue); //setting value
avalue= ui->myCheckBox1->value();    //getting value
//returns true if checkbox is checked and vice versa
ui→chkLetters→isChecked();
```

Setting a max and min range for progress bar and setting a level to be displayed:

```
ui→myProgressBar1→setRange(0,100);    //setting range between 0-100%
ui→myProgressBar1→setValue(35);       //set progress bar to 35%
ui→myProgressBar1→value();            //read value of progress bar
```

Setting and Getting a value from line edit

```
ui→myLineEdit1→setText("foo bar");
ui→myLineEdit→text();                 //Get value from line edit
```

Setting Range for Spinbox and reading value

```
//allowed range of numbers in spinbox
ui→mySpinBox1→setRange(10,30);
int number=ui->spinLength->value();
```

For additional information, sample codes, properties of a Qt classes or widgets, you can always refer to the excellent documentation on http://doc.qt.io/qt-5/index.html

How to get help?

You may ask your questions to TAs or to the instructor. Information regarding the office hours of the TAs and the instructor are available at SUCourse.

WE WILL <u>NOT</u> USE GRADE CHECKER FOR THIS HOMEWORK!

Submit via SUCourse ONLY! Paper, e-mail or any other methods are not acceptable.

The internal clock of SUCourse might be a couple of minutes skewed, so make sure you do <u>not</u> leave the submission to the last minute. In the case of failing to submit your homework on time:

"No successful submission on SUCourse on time = A grade of 0 directly."

What and where to submit (PLEASE READ, IMPORTANT)

You should prepare (or at least test) your program using **Qt Creator v3.0 or above**.

It'd be a good idea to write your name and lastname in the program (as a comment line of course). <u>Do not use any Turkish characters anywhere in your code (not even in comment parts).</u> If your full name is "Duygu Karaoğlan Altop", and if you want to write it as comment; then you must type it as follows:

// Duygu Karaoglan Altop

Submission guidelines are below:

- Name your **<u>Qt project</u>** as:

  "SUCourseUserName_hw6"

  Your SUCourse user name is actually your SUNet username which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SU e-mail address is **atam@sabanciuniv.edu**, then the name of the file to be submitted must be: **"atam_hw6.zip"**.

- Please make sure that the files are the latest versions of your homework program.

- **<u>Compress</u> your Qt project folder including all files through a <u>zip</u> program**. Please use "zip" compression. Do <u>not</u> rename your *.cpp* and *.h* files, keep them as they are, but your project name should follow the rules mentioned above.

- You will receive **no credits** (i) if your compressed zip file does not expand, (ii) if your compressed zip file does not contain the correct files, and/or (iii) if the build process fails.

- Submit your work **through SUCourse only**!

You may visit the office hours if you have any questions regarding submissions.

Plagiarism

Plagiarism is checked by automated tools and we are very capable of detecting such cases. Be careful with that...

Exchange of abstract ideas are totally okay but once you start sharing the code with each other, it is very probable to get caught by plagiarism. So, do NOT send any part of your code to your friends by any means or you might be charged as well, although you have done your homework by yourself. Homeworks are to be done personally and you have to submit your own work. **Cooperation will NOT be counted as an excuse.**

In case of plagiarism, the rules on the Syllabus apply.

Good Luck!
Arsalan Javeed, Tolga Atam, Duygu K. Altop