GENE 121 **Assignment 7** **Fall 2016**

Question 1 and 2 of this assignment will be completed in pairs
Question 3 is to be done **individually**

In this assignment you will be using:

- Functions
- Arrays
- Graphics

Don't forget to include a signed header for your assignment, and to make sure that both students in the group have an electronic copy of the code when it is complete.

Question 1 (Dev-C++)

You are hired to write a program that grades the Pascal Mathematics Competition. The answers of the contest are stored in a file called `pascal2016.txt`. A sample of the file is shown below.

```
D E D C A C A A A C . . .
```

You can assume that there are no more than 30 questions on the contest.

The file `answers2016.txt` stores students' answers to the contest. The first column of the file is the school the student attends, the second is their student ID at that school, followed by their answers to the contest. A sample of such a file is shown below.

```
  WCI 26011 0 E 0 C 0 A . . .
SDCSS 30805 E B 0 C 0 A . . .
  AHS 10566 A D C A A B . . .
   :    :     :    :
```

If a student has a 0 (zero) for an answer to a particular question, it means that they did not answer that question. The contest is graded as follows:

- Each correct answer is worth 5.
- There is no penalty for an incorrect answer.
- Each unanswered question is worth 2, to a maximum of 10 questions.

Your program should do the following:

- Open the files `pascal2016.txt` and `answers2016.txt,` and check whether they are opened properly
- Mark the contest
- Output the students' school, ID and mark to a file called `results.txt` for all students

A sample of the output file is:

```
School      IDs    Marks (%)
 SMCSS     06906      40.0
  SCSS     02584      22.4
   WCI     00365      24.8
     .
     .
```

Submit your code with output for the files given on Learn.

Question 2 (Dev-C++)

Chris has asked the WEEF TAs to take inventory of all the loose resistors in the WEEF Lab. He found four resistors and entered their colour codes in a file, as seen below:

```
4
800
900
1000
500
:
```

Each of the TAs did the same. An example of the next four lines in the file, for a TA who found 328 resistors is shown below:

```
:
328
200
300
800
:
```

The resistor values are all multiples of 100, from 100 to 1000 Ω (i.e. 100, 200, 300, etc…).

Write a main program that

- Opens the file, **`resist.txt`** and verifies that it has opened correctly
- Keeps count of the number of each value of resistor, using an array
- We require at least 275 of each resistor. Print to the file **`report.txt`**, a list of resistors that Chris needs to order, and how many are needed
- Prints to the file the value(s) of the resistor(s) of which we have the most inventory. Note: a tie is possible.

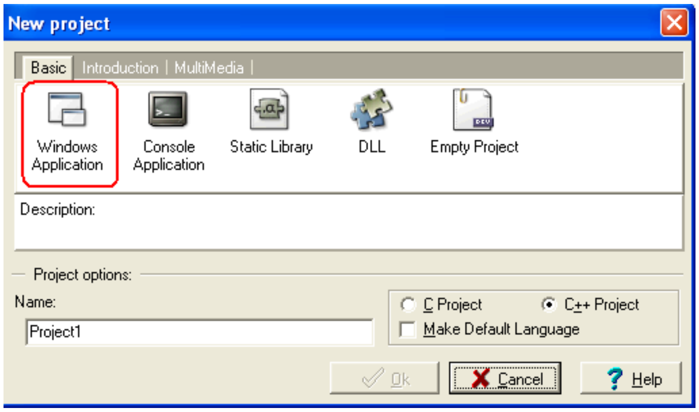
Submit your code with output for the **`resist.txt`** file posted to Learn.

To Use the Graphics Classes

You must set up the compiler differently to get the graphics to run. Follow the steps below:

- Click **File** → **New** → **Project…**

- Choose **Windows Application**. Give the project a suitable name (with no spaces). Click **OK**.



Name: Project1

○ C Project ● C++ Project
□ Make Default Language

- Save the project on your N: drive (or your C: drive if using a laptop, do not use "My Documents").
- A new useless main() will be opened for you.
  - Delete all the code from this file, this is where your program will go
  - Replace the code with:

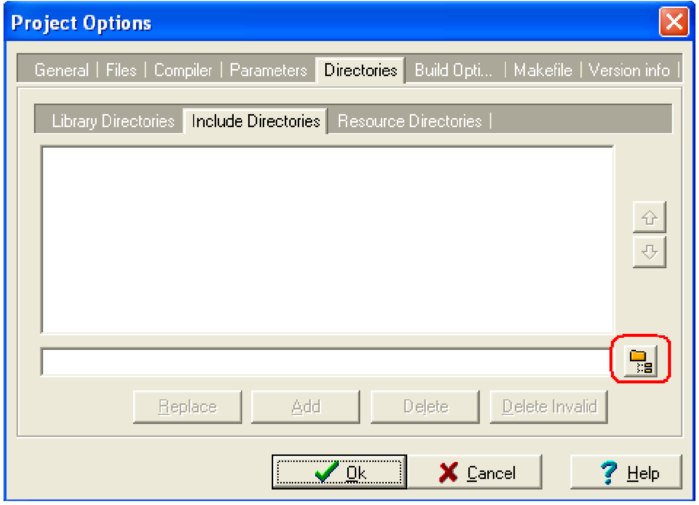
```
// Insert names
#include <cmath>
#include <fstream>

using namespace std;

#include "ccc_win.h"

int ccc_win_main()
{
    // Insert program code here

    return 0;
}
```

- Click on Project -> Project options and select the Compiler tab. For Compiler, use the drop down menu to select TDM-GCC4.9.2 32-bit release

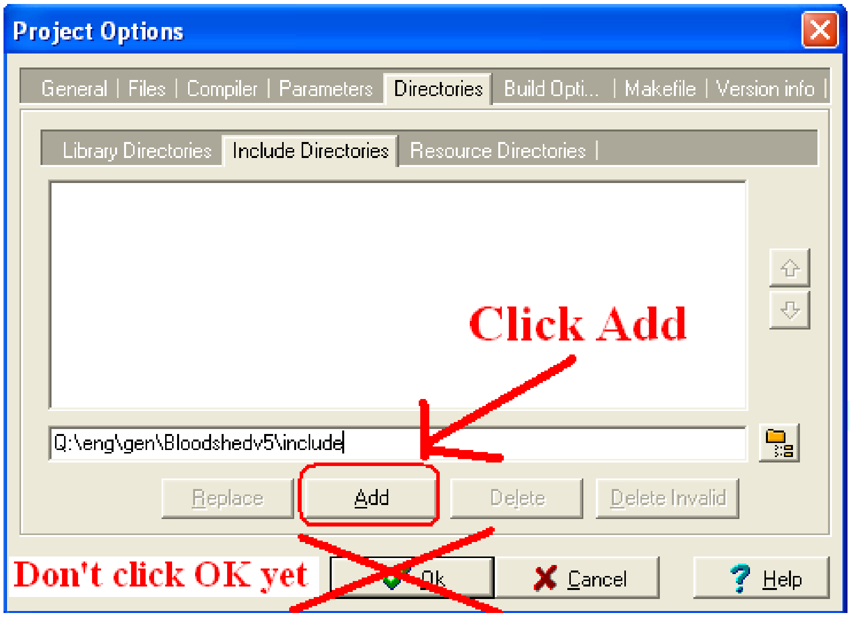
- (This step for laptop users only) Download the graphics files from http://www.horstmann.com/ccc/ccc3e.zip or copy the graphics files from `Q:\eng\gen\Bloodshedv5\include` on one of UW's computers.

- Click on **Project → Project Options.** Select **Directories**, then select **Include Directories**.

- Click on the Browse icon. (There is no name for it, it is just an image of a folder.)


Project Options
General | Files | Compiler | Parameters | Directories | Build Opti... | Makefile | Version info |
Library Directories | Include Directories | Resource Directories |
Replace
Add
Delete
Delete Invalid
Ok
Cancel
Help

- Go to `Q:\eng\gen\Bloodshedv5\include` and select the whole folder. Then click **Add** to add it to the included directories. (If you are using a laptop, go to the folder where you saved the graphics files.)


Project Options
General | Files | Compiler | Parameters | Directories | Build Opti... | Makefile | Version info |
Library Directories | Include Directories | Resource Directories |
Replace
Add
Delete
Delete Invalid
Ok
Cancel
Help

Q:\eng\gen\Bloodshedv5\include

Click Add

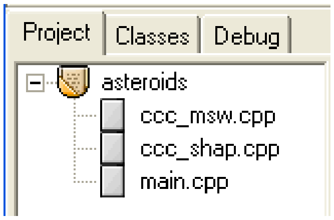
Don't click OK yet

- <u>After the directory has been added</u>, then you can click **Ok**.

- (Skip this step if using a laptop.) Open up **My Computer** outside of Dev-C++. Go to `Q:\eng\gen\Bloodshedv5\include\graphics` and move both files to your working directory. (This is the directory where you have saved your project.)

- Click on **Project → Add to Project**. Add the file `ccc_msw.cpp` to the project. Repeat this step for the file `ccc_shap.cpp`

Your project should now include 3 files: `main.cpp,` `ccc_msw.cpp,` and `ccc_shap.cpp`.



asteroids
ccc_msw.cpp
ccc_shap.cpp
main.cpp

Question 3 – To be completed individually

For Halloween, design something spooky (e.g. black cat, jack-o-lantern, bat, etc.) containing at least one **`Point`**, one **`Circle`** and one **`Line`**.

Write a program that:

- Prompts the user for a number of spooky shapes
- For each spooky shape:
    - Prompts the user to click on a location
    - Draws your spooky design at that location

To capture the output, press Alt and Prt Scrn at the same time. Then paste the output into a Word document or into Paint.

Submit your individual programs and screen shots.