

**Introduction To Computer Programming C++**  
**Mr. Clausen**  
**Program C2A, C2B, C2G**

**Program 2A CelsiusToFahrenheit 20 points**

Write a program that converts a Celsius temperature to Fahrenheit. Save the program as LastNameFirstNameP2A.cpp in your "S:" directory. To see a model for this program look at the source codes for the programs "books.cpp" and "pizza.cpp" in the network directory titled: IntroCompProgFiles. Look in the folder, Other C++ Resources and the folder Other Textbook Examples.

As you type all your programs this year, be sure not to type past the 80-column line in Borland C++ 5.02 for Windows. If you have any statements longer than 80 columns, press the return key to "wrap" the statement around to the next line.

- 1) Type comments at the beginning of the program to display your name and other information just like those used for program 1A.
- 2) Include <conio.h> so you can use getch() to leave your output displayed on the screen until the user presses any key to continue.
- 3) Declare a constant to represent the freezing point of water.

For example: **const double** FREEZING\_POINT\_OF\_WATER = 32.0;

- 4) Inside the **int** main() function, declare variables of type **double** for celsius, and fahrenheit. In these declarations, initialize all of these values to 0.0. These declarations should be on the first few lines after the left curly bracket that begins the main function.
- 5) After the variable declarations, and before the input section, use **cout** statements to display your name and period output just like those used for program 1A

In all of these programs (Program 2A, 2B, and 2C), follow the Input (or **Assignments**), Calculations, and Output organization of your program. Make sure that you include the following comment lines in the **int** main ( ) portion of your program (each comment followed by the appropriate source code):

```
//-----Display My Information-----  
//-----Assignments-----  
//-----Calculations-----  
//-----Output-----
```

- 6) For the Assignment section, assign a value to the double precision floating point variable celsius of 0.0°.
- 7) The calculations section of your program should consist of the comment line and the following formulas:  
The formulas to convert from celsius to fahrenheit is:  
fahrenheit = 1.8 \* celsius + FREEZING\_POINT\_OF\_WATER;
- 8) For the output section of the program use the set precision command found in the source code named SetPrecision.cpp on the network directory named: IntroCompProgFiles. Also in your output portion of your program, echo out the values of the celsius temperature (display it on the screen) before displaying the results of your calculations. Make sure that you use descriptive identifiers for both your variables and your constants.

When you are finished with your program, have tested it thoroughly to make sure that your calculations are correct, and are sure that you don't need to make any changes, then save your program in the "W" network mapping, and the Program 2A folder.

### Program 2B Area Of Circle 25 points

Write a program that calculates the area of a circle. Save the program as LastNameFirstNameP2B.cpp in your "S:" directory. To see a model for this program look at the source codes for the programs "books.cpp" and "pizza.cpp" in the network directory titled: IntroCompProgFiles. Look in the folder, Other C++ Resources and the folder Other Textbook Examples.

As you type all your programs this year, be sure not to type past the 80-column line in Borland C++ 5.02 for Windows. If you have any statements longer than 80 columns, press the return key to "wrap" the statement around to the next line.

- 1) Type comments at the beginning of the program to display your name and other information just like those used for program 1A.
- 2) Include <conio.h> so you can use getch() to leave your output displayed on the screen until the user presses any key to continue.
- 3) Declare a constant of type **double** for PI.

For example: **const double** PI = 3.14159;

- 4) Inside the **int** main() function, declare variables of type **double** for radius and area. In these declarations, initialize all of these values to 0.0. These declarations should be on the first few lines after the left curly bracket that begins the main function.
- 5) After the variable declarations, and before the input section, use **cout** statements to display your name and period output just like those used for program 1A

In all of these programs (Program 2A, 2B, and 2C), follow the **Input**, Calculations, and Output organization of your program. Make sure that you include the following comment lines in the **int** main ( ) portion of your program (each comment followed by the appropriate source code):

```
//-----Display My Information-----  
//-----Input-----  
//-----Calculations-----  
//-----Output-----
```

- 6) For the Input section, ask the user to enter the value for the radius. Make your program user friendly by prompting them for this value.
- 7) The calculations section of your program should consist of the comment line and the following formula: The formula to calculate the area of a circle is:

area = PI \* radius \* radius;

- 8) For the output section of the program use the set precision command found in the source code named SetPrecision.cpp on the network directory named: IntroCompProgFiles. Also in your output portion of your program, echo out the value of the radius before displaying the results of your calculations: the answers to the area of the circle. Make sure that you use descriptive identifiers for your variables and your constants.

When you are finished with your program, have tested it thoroughly to make sure that your calculations are correct, and are sure that you don't need to make any changes, then save your program in the "W" network mapping, and the Program 2B folder.

**Program 2G Harry or Harriette (25 points)**

Write a graphics program that draws a face.

1) Open the source code for the program "GraphicsTemplate.cpp" in the network directory titled: IntroCompProgFiles. Look in the folder, Other C++ Resources and the folder Graphics Save the program as LastNameFirstNameP2G.cpp in your "S:" directory.

2) Don't forget that you need to create a DOS project with the proper settings for your program to work.

Here is the link to the notes on how to set up the project:

[http://www.lcusd.net/lchs/dclausen/intro\\_programming/Lectures/Borland Graphics Project.pdf](http://www.lcusd.net/lchs/dclausen/intro_programming/Lectures/Borland_Graphics_Project.pdf)

The Name of the project should be your ID number, then PC2G. For example: 9999PC2G.ide

As you type all your programs this year, be sure not to type past the 80-column line in Borland C++ 5.02 for Windows. If you have any statements longer than 80 columns, press the return key to "wrap" the statement around to the next line.

3) Type comments at the beginning of the program to display your name and other information just like those used for program 1A.

4) Include <conio.h> so you can use getch() to leave your output displayed on the screen until the user presses any key to continue. Include <graphics.h> so you can draw your graphics commands.

5) Inside the **int** main() function, add your commands to the program to complete the assignment. Use the graphics commands for: Line, Rectangle, Circle, Ellipse, Arcs, and Points (please make sure to use all of them). Draw a picture of a face (the face can be a square, rectangle, circle, etc.) including eyes, nose, mouth, etc. Have the face fill the screen (no little faces here) and add as much detail as you can.

6) You need to separate the commands that draw each part of the face from the other commands with comment lines that tell me which part of the face you are drawing. For example:

```
//-----Left Eye-----
```

```
//-----Right Eye-----
```

```
//-----Mouth-----
```

and all of the other parts of the face.