

Honors Computer Science C++

Mr. Clausen

Program 3A, 3B, 3C

Program 3A Cone Heads (25 points)

Write a program to calculate the volume and surface area of a right circular cone. Allow the user to enter values into variables for the radius and the height. Save the program as LastNameFirstNameP3A.cpp in your "S:" directory. To see a model for this program look at the source code for the program "pizza.cpp" in the network directory titled: HonorsCompSciFiles. Look in the folder, Text Book Programs and Ch3.

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 2A. **Be sure to change the program name, program number, and program description.**

- 2) Include the preprocessor directives:

```
#include <iostream.h>      //necessary for cout and cin
#include <iomanip.h>        //necessary for setprecision
#include <conio.h>          //necessary for getch()
#include <math.h>           //necessary for sqrt
```

- 3) Next declare all of the constants necessary for this program. Declare a constant of type **double** for $PI = 3.14159$, and declare a constant of type **double** named `ONE_THIRD = 1.0/3.0`.

- 4) Declare the **int** `main()` function here.

- 5) Declare all of the variables necessary for this program. You will need variables of type **double** for radius, height, volume, and surface_area. Make sure that you use descriptive variable names to create readable, self-documenting code.

- 6) Type the following comment:

```
//-----Display My Information-----
```

Follow this comment with cout statements to display your name and period output just like those used for program 2A. **Be sure to change the program name, and program number.**

- 7) For the Input section, type the following comment:

```
//-----Input-----
```

Ask the user to enter the radius and height of the cone. Make your program user friendly by using “cout” (without an endl) statements prompting the user for these.

- 8) For the Calculations section, type the following comment:

```
//-----Calculations-----
```

The formulas are:

```
volume = ONE_THIRD * PI * radius * radius * height
```

```
surface_area = PI * radius * (radius + sqrt(radius * radius + height * height))
```

- 9) For the Output section, type the following comment:

```
//-----Output-----
```

For the output portion of your program, echo out the values of the radius and height as well as the answers to the volume and surface area. Make this output easy to read and formatted nicely.

Use blank lines to separate each of the program sections listed in all of the steps above. 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, in the Program 3A folder.

Program 3B Identity Crisis and Lucky Numbers (25 points)

Write a program to practice many of the language features that we learned in Chapter 3. Save the program as LastNameFirstNameP3B.cpp in your “S:” directory. To see a model for this program look at the source code for the program “pizza.cpp” in the network directory titled: HonorsCompSciCFiles. Look in the folder, Text Book Programs and Ch3. **Also look at the program named “Random.cpp” in the network directory titled: HonorsCompSciCFiles. Look in the folder, named “Random”. There are comments in this program to tell you which lines of code to copy and paste into your program 3B.**

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 2A. **Be sure to change the program name, program number, and program description.**
- 2) Include the preprocessor directives:

```
#include <iostream.h>          //necessary for cout and cin
```

```

#include <iomanip.h>           //necessary for setprecision
#include <conio.h>            //necessary for getch()
#include <math.h>             //necessary for sqrt and pow
#include <stdlib.h>          //necessary for random numbers
#include <time.h>            //necessary for the clock to seed the random numbers
#include "apstring.cpp"      //necessary for String variables

```

- 3) Next declare all of the constants necessary for this program. Declare a constant of type **int** for LOW = 1, and declare a constant of type **int** named HIGH = 100.
- 4) Declare the **int** main() function here.
- 5) Declare all of the variables necessary for this program. You will need variables of type **double** for base, id_square_root, base_to_power, variables of type **int** for id_number, exponent, ascii_code, random_number, and name_length, a variable of type **char** for letter, variables of type **apstring** for consume_newline, and full_name, and a variable of type time_t called seconds (time_t seconds;). Make sure that you use descriptive variable names to create readable, self-documenting code.
- 6) Type the following comment:

```
//-----Display My Information-----
```

Follow this comment with cout statements to display your name and period output just like those used for program 2A. **Be sure to change the program name, and program number.**
- 7) For the Input section, type the following comment:

```
//-----Input-----
```

Ask the user to enter their ID number and assign this to the variable id_number. Ask the user to enter their full name (first name space last name) and assign this to the variable full_name (**Do not change the order of these first two inputs or you will lose points on your program**). Use the apstring function getline to enter this information into a variable of type apstring. Remember that you will need to “consume” the new line character before asking for the name in order for this to work correctly. Ask the user for a positive real number between 1.0 and 25.0 and assign this to the variable base. Ask the user to enter a positive integer between 1 and 5 and assign this to the variable exponent. Ask the user to enter a letter of the alphabet (either upper or lower case) and assign this to the variable letter. Make your program user friendly by using “cout” (without an endl) statements prompting the user for these.
- 8) For the Calculations section, type the following comment:

```
//-----Calculations-----
```

For the calculations section assign the results of the calculations to the remaining appropriate variables that you declared above. Calculate how many characters are in the name that the user entered, including the space. You will need to use the apstring function length to do this. Calculate the square root of the ID number, raise the base to the power, and explicitly cast the character variable to an integer variable. Next add the

two lines of code to “randomize” your random number. Finally, calculate the random number as illustrated in the sample program “Random.cpp”.

- 9) For the Output section, type the following comment:

```
//-----Output-----
```

For the output portion of your program, set the precision to 2 decimal places. Display the users’ ID number, full name, the length of their full name, the square root of their ID number, the base, exponent, the value of the base raised to the exponent, the character the user entered, the ASCII code of that character, and the random number. Make this output easy to read and formatted nicely.

Use blank lines to separate each of the program sections listed in all of the steps above. 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, in the Program 3B folder.

PROGRAM 3C Full Names (25 points)

Write a program to practice using the `apstring` class and it’s functions. Save the program as `LastNameFirstNameP3C.cpp` in your “S:” directory. To see a model for this program look at the source code for the program “pizza.cpp” in the network directory titled: `HonorsCompSciCFiles`. Look in the folder, `Text Book Programs` and `Ch3`.

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 2A. **Be sure to change the program name, program number, and program description.**

- 2) Include the preprocessor directives:

```
#include <iostream.h>          //necessary for cout and cin
#include <conio.h>              //necessary for getch()
#include “apstring.cpp”        //necessary for String variables
```

- 3) There are no constants necessary for this program.

- 4) Declare the `int` `main()` function here.

- 5) Declare all of the variables necessary for this program. You will need variables of type `int` for `first_space_location`, `second_space_location`, `full_name_length`, and `temp_name_length`, and variables of type `apstring` for `full_name`, `first_name`, `middle_name`, `last_name`, and `temp_name`. Make sure that you use descriptive variable

names to create readable, self-documenting code.

- 6) Type the following comment:

```
//-----Display My Information-----
```

Follow this comment with cout statements to display your name and period output just like those used for program 2A. **Be sure to change the program name, and program number.**

- 7) For the Input section, type the following comment:

```
//-----Input-----
```

Ask the user to enter their full name (first name, a space, middle name, a space, and last name). You will need to use a getline statement. Make your program user friendly by using a “cout” (without an endl) statement prompting the user for this.

- 8) For the Calculations section, type the following comment:

```
//-----Calculations -----
```

Your job is to separate this full name into three separate variables: one for the first name, one for the middle name, and a third for the last name. You will need to use the **find**, **length**, and **substr** functions built in to the **apstring** class. Here’s a hint, the spaces tell you where the first, middle and last names can be found.

- 9) For the Output section, type the following comment:

```
//-----Output-----
```

For the output portion of your program, display the full name as well as the first name, middle name, and last name. Make this output easy to read and formatted nicely.

Use blank lines to separate each of the program sections listed in all of the steps above. 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, in the Program 3C folder.