CSC 111 Programming Assignment 3

Programming instructions

Assignment 3 consists of two parts. Each part requires a separate C program in a separate .c source file. Upload the syntactically and semantically correct C source files (i.e., .c files) using your CourseSpaces drop box (cf. instructions below).

Part I

You are to create a small library of syntactically and semantically correct C functions with appropriate parameters where your library will have one function to compute each of the following:

1. the area of a circle with radius ***r***—use constant `M_PI` for π which is defined in `<math.h>`; thus, you need the following include directive: `#include <math.h>` for this program.
2. the perimeter of a circle with radius ***r***
3. the volume of a sphere with radius ***r***
4. the surface area of a sphere with radius ***r***
5. the volume of a cylinder with base radius ***r*** and height ***h***—call function 1 to implement this function
6. the surface area of a cylinder with base radius ***r*** and height ***h*** including the top and bottom of the cylinder)—call functions 1 and 2 to implement this function
7. the volume of a cone with base radius ***r*** and height ***h***—call function 1 to implement this function
8. the surface area of a cone with base radius ***r*** and height ***h*** including the bottom of the cone—call function 1 to implement this function

For example, (1), (2) and (5) above can be implemented as follows:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define RadiusRange1 (7.3)
#define RadiusInc (0.7)
#define RadiusRange2 (13.2)
#define CylinderHeight1 (8.2)
#define CylinderHeightInc (1.5)

// Computes area of a circle with radius r
double areaCircle(double r){
   return M_PI*r*r;
}/*areaCircle*/

// Computes perimeter of a circle with radius r
double periCircle(double r){
   return M_PI*(r+r);
}/*periCircle*/

// Computes volume of a cylinder with base radius r and height h
double volCylinder(double r, double h){
   double areaCi = areaCircle(r);
   double volCy = h*areaCi;
   return volCy;
```

```
}/*volCylinder*/

int main(void){
   double radius = 6.2;
   double area = areaCircle(radius);
   double height = 8.1;
   double vol = volCylinder(radius, height);
   printf("Area of circle with radius %g is %.2f\n", radius, area);
   printf("Volume of cylinder with radius %g and height %g is %.2f\n",
   radius, height, vol);
   return EXIT_SUCCESS;
}/*main*/
```

After writing all eight functions, modify the `main()` function above. In a loop, iterate the `radius` from `7.3` to `12.9` and cylinder `height` from `8.2` to `20.2`. Call the functions `areaCircle()`, `periCircle()`, `volCylinder()`to produce the following output. The numbers in the first row below help in calculating the field widths required for the output. The numbers are required to ease grading.

```
01234567 01234567 01234567 01234567 01234567
  Radius CircArea CircPeri   CylHei   CylVol
     7.3   167.42    45.87      8.2  1372.81
     8.0   201.06    50.27      9.7  1950.30
     8.7   237.79    54.66     11.2  2663.22
     9.4   277.59    59.06     12.7  3525.41
    10.1   320.47    63.46     14.2  4550.73
    10.8   366.44    67.86     15.7  5753.04
    11.5   415.48    72.26     17.2  7146.18
    12.2   467.59    76.65     18.7  8744.02
    12.9   522.79    81.05     20.2 10560.41
```

Part II

Write a C program that implements a calculator for a right circular cone as follows. Three characteristic variables of a cone are its *base radius* `r`, its *height* `h` and its *volume* `v`. Given the values of two of these variables, the calculator computes the appropriate value for the third variable. Use `double` for all floating-point variables.

Detailed instructions: In an infinite loop generate random numbers in the range 0 to 3 using the C library function `rand()` defined in `stdlib.h`. Then depending on the random number generated, do one of the following:

0 — exit program, but only after at least 5 cone calculations
1 — compute `v` using `r` and `h` which are input on the console by the user
2 — compute `h` using `r` and `v` which are input on the console by the user
3 — compute `r` using `h` and `v` which are input on the console by the user

From the console, read the floating-point values into `double` variables using the `%lf` format specifier and the `scanf()` function, which is defined in `stdio.h`. Ignore the warnings produced by the compiler when using `scanf()` and `ran()`. Use the function `sqrt()` defined in `math.h` to compute `r`.

A console below illustrates sample interaction between the program and the user. Note that CLions/MinGW under Windows will repeat the values the user entered. Count the iterations (i.e., "3 ConCalc: …").

```
Welcome to my cone calculator
Enter r, h:3 4
1 ConeCalc: r = 3.00  h = 4.00  v = 37.70
Enter h, v:5 6
2 ConeCalc: r = 1.07  h = 5.00  v = 6.00
Enter r, v:7 8
3 ConeCalc: r = 7.00  h = 0.16  v = 8.00
Enter r, h:9 10
4 ConeCalc: r = 9.00  h = 10.00  v = 848.23
Enter r, v:11 12
5 ConeCalc: r = 11.00  h = 0.09  v = 12.00
Enter r, v:13 14
6 ConeCalc: r = 13.00  h = 0.08  v = 14.00
Enter r, v:15 16
7 ConeCalc: r = 15.00  h = 0.07  v = 16.00
Bye
```

General instructions

Write the programs using pencil and paper. Edit the programs using the CLion environment in ECS 242 or on your personal workstation including a comment at the top of each file as follows:

```
Name: Polar Bear – replace with your name
UVicID: V00123456 – replace with your UVicID
Date: 2017/10/02 – replace with the date you write the program
Assignment: A3
File names: V00123456A3P1.c replace 00123456 with your UVicID
Description: replace with a description what the program does
```

You need to submit two files named as follows where **V00123456** is replaced with your UVicID:

- **V00123456A3P1.c ⟷ P1 refers to Part 1**
- **V00123456A3P2.c ⟷ P2 refers to Part 2**

If the programs are syntactically and semantically correct submit the C source files to your ***CourseSpaces Drop Box*** using the submission instructions below.

Assignment submission instructions

CSC 111 assignments will only be accepted electronically through the assignment page on the CSC 111 course Moodle site using the ***Drop Box***. Your submission will consist of a ***two C source files***. If your student ID is ***V00123456*** and the program is for ***Assignment 3***, your C source files for the three different parts must be named as follows: **V00123456A3P1.c** and **V00123456A3P2.c** In addition, your ***full name, student ID,*** and ***Assignment name*** (e.g., Assignment 3) must appear in a ***comment section at the beginning of your C programs***. Please submit only the source files and not the executable files. Submissions that do not follow these guidelines may not be graded. Since this assignment requires more than one source file, *CourseSpaces* will allow you to submit two different files. Until the assignment due date, you may change your submissions by deleting and resubmitting your source files multiple times. After the due date, no submissions will be accepted. When grading is complete, your assignment mark and comments will appear in the ***CourseSpaces Gradebook.***