COP 3014: Fall 2021
Homework 2

Total Points: 100
Due: Wednesday, 09/22/2021 11:59:00 PM EDT

1 Objective

The purpose of this assignment is to

- Practice using CLion to create, edit, compile, debug, and run a c++ program.
- Use the `iostream` library to perform console input and output.
- Use variables and arithmetic operators in C++ to do basic math.
- Familiarize yourself with selection statements (if . . . else and switch).
- Use fundamental algorithmic error checking (checking the validity of user input).

This assignment requires you to submit 2 files on Canvas. Please do so in a SINGLE submission. Canvas allows you to turn in multiple flies in one submission. Once you have uploaded your first file, click "Attach Another File" to upload your second file.

Turn in your 2 files `slices.cpp` and `timer.cpp` to Canvas.

2 Pizza "Ordering"

Incidental Fish 39, inspired by Bubble Bass' calculations has professed his desire to create a system to rank Krusty Krab pizzas based on area per slice. He will get the radii of the 3 pizza sizes that the Krusty Krab offers, not necessarily in order, and the number of slices in each size, and rank them in ascending order of area per slice. While Incidental 39 is good at taking advantage of situations ad-hoc, he is not that great at math or remembering things. So, he enlists your help.

Write a C++ program to get pizza radii and number of slices per pizza for 3 sizes of pizza as input and rank the pizzas in ascending order of area per slice.

Call this program `slices.cpp`

Specifications

1. Prompt the user and read in 3 pizza radii. Store them in 3 variables. These need not be in the small - medium - large order. (5 points)
2. Prompt the user and read in the number of slices for each of the 3 pizza sizes entered. The number of slices will correspond to the pizza size, ie, the first number of slices is for the first pizza size. (5 points)
3. If the user entered values become the same 3 areas per slice, print an error message. (10 points)

4. Use `if ...else` statements, in whatever combination required, to print the pizza numbers and their areas per slice out in ascending order. (25 points)

5. Add comments wherever appropriate to explain your logic. (5 points)

6. You can assume that the user will only enter the proper type of data for the values. Positive floating-point numbers for the radii and positive integers for the number of slices. No characters or text.

7. You may use 3.14 as the value for Π

8. Print the values accurate to 3 digits after the decimal point.

Sample Run 1

The underlined text is the user input.

```
Enter the 3 radii values:  12.5
18.2
26.4
Enter the number of slices for each:  8
12
14
Pizza 1:  61.328, Pizza2:  86.674, Pizza3:  156.318
```

Sample Run 2

The underlined text is the user input.

```
Enter the 3 radii values:  19.378
4.129
6.2716
Enter the number of slices for each:  12
2
3
Pizza 2:  26.766, Pizza3:  41.169, Pizza1:  98.258
```

Sample Run 3

The underlined text is the user input.

```
   Enter the 3 radii values:  30
20
10
Enter the number of slices for each:  9
4
1
Error.  You entered the same area per slice 3 times.
```

3 Problem 2 - Timing birds traversing a distance

Arthur, King of the Britons, has tasked you with figuring out the amount of time it would take his faithful messenger birds to carry messages (which may or may not be in coconuts) across certain

distances. His assistant, Patsy, has provided you with a table that contains the speed of the birds (in meters per second), and the letter with which the birds will be referred to. Use the table to calculate the times.

Call this file `timer.cpp`

Bird	**Speed (m/s)**	**Reference Letter**
African Swallow	13.64	A
African Swallow + Coconut	11.91	C
European Swallow	12.32	E
European Swallow + Coconut	10.95	U
Barn Swallow	10.07	B
Barn Swallow + Coconut	9.86	W

Specifications

1. Print a menu for the user. In the menu, display the birds and their reference letters. (7 points).
2. Prompt the user the enter the bird in question. Read in the reference letter. (5 points)
3. Prompt the user to enter the distance (in kilometers) and read in the value (5 points).
4. Use a `switch` statement on the reference letter to figure out the speed. (15 points).
5. If the reference letter entered doesn't match any bird, print an error message (5 points)
6. Calculate the time taken by the bird and print it, accurate to 4 digits after the decimal point. (8 points).
7. Add comments wherever appropriate to explain your logic. (5 points)
8. You may assume that the user will only enter the proper type of data, an uppercase character for the reference letter and a floating point number for the distance.
9. 1 kilometer = 1000 meters
10. speed = $\frac{distance}{time}$

Sample Run 1

The underlined text is the user input.

```
African Swallow - A
African Swallow + Coconut - C
European Swallow - E
European Swallow + Coconut - U
Barn Swallow - B
Barn Swallow + Coconut - W
Enter the bird:  C
Enter the distance (in kilometers):  12.5
The time taken is 1049.5382 seconds.
```

Sample Run 2

The underlined text is the user input.

```
African Swallow - A
African Swallow + Coconut - C
European Swallow - E
European Swallow + Coconut - U
Barn Swallow - B
Barn Swallow + Coconut - W
Enter the bird:  X
Enter the distance (in kilometers):  19
Error:  Invalid reference letter for the bird.
```

4 General Requirements

1. Include the header comment with your name and other information on the top of your files.
2. Please make sure you're only using the concepts already discussed in class. Please restrict yourself to variables, operators and selection statements. Using loops, arrays, C++ 11 and up features or anything more advanced will result in a loss of 10 points.
3. Each program is worth 50 points.
4. If we have listed a specification and allocated point for it, you will lose points if that particular item is missing from your code, even if it is trivial.
5. No global variables (variables outside of main())
6. You may not use the `auto` keyword.
7. **This is individual work. You may NOT collaborate with other students in the course, former students, hire tutors to "help", copy solutions off the internet, or use pay-for solution websites, including but not limited to Chegg, CourseHero, WiseAnt, Bartelby, assorted Social Media groups, etc.) Doing so is a violation of the Academic Honor Code.**
8. All input and output must be done with streams, using the library `iostream`
9. You may only use the `iostream` and `iomanip` libraries (you do not need any others for these tasks). Use of other libraries would result in a loss of 10 points per library.
10. NO C style printing is permitted. (Aka, don't use printf). Use cout if you need to print to the screen.
11. When you write source code, it should be readable and well-documented (comments).
12. Make sure you either develop with or test with CLion (to be sure it reports no compile errors or warnings) before you submit the program.
13. Testing your program thoroughly is a part of writing good code. We give you sample runs to make sure you match our output requirements and to get a general idea of how we would test your code. Matching your outputs for JUST the sample runs is not a guarantee of a 100. We have several extensive test cases.
14. Please make sure you've compiled and run your program before you turn it in. Compilation errors can be quite costly. We take 5 points off per compiler error for the first 9 errors. The 10th compiler error will result in a grade of 0.

15. Only a file turned in through Canvas counts as a submission. A file on you computer, even if it hasn't been edited after the deadline, does not count.

16. The student is responsible for making sure they have turned in the right file(s). We will not accept any excuses about inadvertently modifying or deleting files, or turning in the wrong files.

17. **Program submissions** should be done through the Canvas class page, under the assignments tab (if it's not there yet I'll create it soon.) Do not send program submissions through e-mail – e-mail attachments will not be accepted as valid submissions.

18. The ONLY files you will submit via Canvas are `slices.cpp` and `timer.cpp`

19. **General Advice** - always keep an untouched copy of your finished homework files in your email. These files will have a time-stamp which will show when they were last worked on and will serve as a backup in case you ever have legitimate problems with submitting files through Canvas. Do this for ALL programs.