**CIS 1275 C++ Programming Part I Name:**

**Program 5 Fish Showers**

# References and Pointers in Functions Due: Wednesday, November 4, 2015

**Turn in Requirements:**

1. **5 pts.** Name your project LastnameP5, such as NelsonP5.
2. **10 pts.**  Please print your source code (Name your files **Driver.cpp, FishShowerFunctions.h** and **FishShowerFunctions.cpp**), attach it to this page to turn in.
3. **5 pts.** Remove the .sdf and debug files and then upload your zipped project through Blackboard

**Program Requirements:**

1. **5 pts.**  Write your name, email address and file name at the t op of your source code in a comment.
2. **5 pts.** Use cout statements to write your name, program title, and program objective to the screen so that it is the first thing I see when your program runs. This is your course header.
3. **5 pts.** Add comments as appropriate. Be sure that your program output is neatly presented to the user

**20 pts Test case write up (see below). Due on Monday, March 3, 2014 by 4:50pm**

**80 pts Program, includes the points listed above**

The C++ Fish Shower provides the fish in your pond a shower they can use at any time. The Fish Shower sprays a cone of water into the air causing a rain-like effect on the surface of your pond, as well as a splashy-splashy sound. It is important that the pond is fitted with the correct shower for the pond owner and the pond fish’s enjoyment.

Two examples of fish showers may include the “Shebunkin Sprayer” and “Koi Rain Head.” You can look up pond fish on the web and come up with appropriate names for shower types. The recommended pond size range for your shower types will be:

* 200 – 400 gallons
* 300 - 500 gallons
* 400 – 1000 gallons
* 900 – 5000 gallons
* 3000 – 10000 gallons

For example, the Shebunkin Sprayer is optimum for small ponds between 200 and 400 gallons. The Koi Rain Head is an excellent choice for ponds between 300 and 500 gallons. The Fish Shower Types and the Optimum Pond Minimum and Maximum sizes are stored in three parallel vectors.

Begin your program in main by declaring four vectors, one for the FishShowerType, one for the minimum pond size for that FishShowerType, one for the maximum pond size for that FishShowerType, and a vector to hold recommendations if the user has selected a FishShower that will not work. The first three three vectors will have five elements, corresponding to the five FishShowerTypes. Call the function *FillVectors*, passing in references to the first three vectors. The recommendation vector should remain empty.

Present your course header and any additional information concerning the Fish Showers. Begin a do-while or while loop and call the function *AskFishShowerTypes*, which presents the user with a menu of five Fish Shower Models. It should pass in a reference to the vector FishShowerType and a pointer to the index, which refers to the index of the FishShowerType chosen.

Next, determine the number of gallons in the user’s pond. Call *CalcPondVol* which asks the pond shape (circular or rectangular) and depth and determines the pond volume. Assume a constant depth. The argument in this function call is a pointer to the total gallons in the pond.

Finally, pass the user’s desired FishShowerType index, the pond volume and references to the four vectors into the *ValidateFishShower* function. The return is a bool, and there are three possibilities:

1. True, indicating that the pond volume is in the range of the selected FishShowerType
2. False, there could be two possibilities: 1) the pond volume is NOT in the range of the selected FishShowerType OR that the pond volume is not in ANY of the ranges of the FishShowerTypes

If the case is 2, above, your program needs to determine if there is no possible shower, or find the perfect FishShowerType for that pond. In the function, check the four other ranges to search for an appropriate selection. Since the ranges overlap, you may get more than one appropriate range. As you search, each time you find a range, add that index to the recommendations vector.

Finally, in main, check the return from *ValidateFishShowe*r. If the return is true, call *WriteInfo*, passing in the name of the selected FishShowerType and the pond size. If the return is false, call the **overloaded** function *WriteInfo,* which passes in the name of the selected FishShowerType, the pond size and references to the FishShowerType vector and the recommendation vector.

Ask is the user wants to play again. If not, present a good-bye message. All functions are called from main.

The functions you will need are:

|  |  |  |  |
| --- | --- | --- | --- |
| **Return Type** | **Function Name** | **Argument List** | **Description** |
| **void** | WriteHeader | ( ) | Display the course header and any additional information to the user. |
| **void** | FillVectors | (vector<string> &type, vector<int> &minGal, vector<int> &maxGal) | Load the vectors with five elements each |
| **void** | AskFishShower | (vector<string> &type,  int \*pIndex) | Display a menu and allow the user to select a FishShower type |
| **void** | CalcPondVol | (int \*pGal) | Ask the user for the pond dimensions and calculate the volume of the pond in gallons. |
| **bool** | VallidateFishShower | (int index, int pondVol, vector<int> &minGals, vector<int> &maxGals, vector<int> &recommendations) | Check that the pond volume corresponding to the selected Fish Shower works. If not, suggest other Fish Showers. |
| **void** | WriteInfo | (string showerType, int gallons) | Display result if selection is valid |
| **void** | WriteInfo | (string showerType, vector<string> &showerTypes, vector<int> &recommendations, int gallons) | Display result including recommendation of a better Fish ShowerType. Report if the pond volume is outside all of the ranges. |

**Test case write up:**

Please see the Test Case assignment sheet.