src/functions.cpp

```cpp
// Title:  functions.cpp
// Desc:   Definds following functions: getInput(), calculateLakeArea(),
calculateLakeVolume(),
//         calculateFishStock(), and calculateMaxLicenses()
// Name:   An Tran

//TODO - Add the appropriate file header here
#include "functions.h"
// TODO - Add any needed #include statements here
#include <iostream>

// TODO - Implemented the following functions. Refer to the assignment README as
needed

int getInput() {
/*********************YOUR CODE BELOW THIS LINE**********************/
int userInput{0};

std::cin >> userInput;

return userInput;
/*********************YOUR CODE ABOVE THIS LINE**********************/
}

float calculateLakeArea() {
/*********************YOUR CODE BELOW THIS LINE**********************/
float lakeArea{0.0};
float hValue{0.0};
hValue = 200.0; // Width of each segment
float depth0{0.0}, depth1{0.0}, depth2{0.0}, depth3{0.0}, depth4{0.0}, depth5{0.0},
      depth6{0.0}, depth7{0.0}, depth8{0.0}; // Depth at each point

//This section of getting user input for the depths could be done more modularly
std::cout << "Enter depth for P0: ";
depth0 = getInput();
std::cout << "Enter depth for P1: ";
depth1 = getInput();
std::cout << "Enter depth for P2: ";
depth2 = getInput();
std::cout << "Enter depth for P3: ";
depth3 = getInput();
std::cout << "Enter depth for P4: ";
depth4 = getInput();
std::cout << "Enter depth for P5: ";
depth5 = getInput();
std::cout << "Enter depth for P6: ";
depth6 = getInput();
std::cout << "Enter depth for P7: ";
depth7 = getInput();
std::cout << "Enter depth for P8: ";
depth8 = getInput();


//Find area using Simpson's Rule
// Area = (h/3) * (y0 + 4 * (y1 + y3 + ... + yn-1) + 2 * (y2 + y4 + ... + yn-2 + yn))
lakeArea = (hValue / 3.0) * (depth0 + (4 * depth1) + (2 * depth2) + (4 * depth3) + (2
* depth4)
            + (4 * depth5) + (2 * depth6) + (4 * depth7) + depth8);

return lakeArea;
/**********************YOUR CODE ABOVE THIS LINE************************/
}


float calculateLakeVolume(float areaOfLake) {
    /**********************YOUR CODE BELOW THIS LINE************************/
float lakeVolume{0};

// volume = area * depth
lakeVolume = areaOfLake * 20; // 20 is the average lake depth

return lakeVolume;
    /**********************YOUR CODE ABOVE THIS LINE************************/
}

int calculateFishStock(float volumeOfLake) {
    /**********************YOUR CODE BELOW THIS LINE************************/
int fishStock{0};

// 1 fish per 100 cubic feet
fishStock = volumeOfLake / 1000;

return fishStock;
    /**********************YOUR CODE ABOVE THIS LINE************************/
}

int calculateMaxLicenses(int fishStock) {
    /**********************YOUR CODE BELOW THIS LINE************************/
int maxLicenses{0};
float availableFishStock{0};

//25% of the original fish population must remain at the end of the season
availableFishStock = fishStock * 0.75;

//Average catch is 20 fish per license
maxLicenses = availableFishStock / 20;

return maxLicenses;
    /**********************YOUR CODE ABOVE THIS LINE************************/
}
```