Exercise 3-2: Using Libraries

Part A:

* abs();
* At least 1 int/double/float type
* The absolute value of the user generated int/double/float is returned
* Code:

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double num;

cout << "Enter a number: ";

cin >> num;

cout << "The absolute value for " << num << " is " << abs(num) <<

endl << endl;

system("pause");

return 0;

}

Part B:

* pow(x, y);
* 2 int/double/float types
* The function returns the value of x raised to the power of y
* Code:

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double numX;

double numY;

cout << "Enter base number: ";

cin >> numX;

cout << "Enter exponent: ";

cin >> numY;

cout << "\n" << numX << " to the power of " << numY << " is " <<

pow(numX, numY) << endl << endl;

system("pause");

return 0;

}

Exercise 3-4: Output Formatting

[Attached source.cpp, OUT, and racedat.txt files with assignment upload section]

#include <iostream>

#include <iomanip>

#include <string>

#include <fstream>

using namespace std;

int main()

{

ifstream inFile;

ofstream outFile;

char athlete1Letter;

double athlete1Time1, athlete1Time2, athlete1Time3, athlete1TotalTime;

char athlete2Letter;

double athlete2Time1, athlete2Time2, athlete2Time3, athlete2TotalTime;

char athlete3Letter;

double athlete3Time1, athlete3Time2, athlete3Time3, athlete3TotalTime;

char athlete4Letter;

double athlete4Time1, athlete4Time2, athlete4Time3, athlete4TotalTime;

double avgTotalTime;

inFile.open("racedat.txt");

outFile.open("racedat.out");

cout << "Creating file with race data info...\n";

outFile << "Enter athlete identifier followed by the 3 completion times for running, swimming, and biking.\n";

outFile << "Do this 4 times for each of the 4 athletes with time entered in minutes...\n";

inFile >> athlete1Letter >> athlete1Time1 >> athlete1Time2 >> athlete1Time3;

inFile >> athlete2Letter >> athlete2Time1 >> athlete2Time2 >> athlete2Time3;

inFile >> athlete3Letter >> athlete3Time1 >> athlete3Time2 >> athlete3Time3;

inFile >> athlete4Letter >> athlete4Time1 >> athlete4Time2 >> athlete4Time3;

athlete1TotalTime = athlete1Time1 + athlete1Time2 + athlete1Time3;

athlete2TotalTime = athlete2Time1 + athlete2Time2 + athlete2Time3;

athlete3TotalTime = athlete3Time1 + athlete3Time2 + athlete3Time3;

athlete4TotalTime = athlete4Time1 + athlete4Time2 + athlete4Time3;

avgTotalTime = (athlete1TotalTime + athlete2TotalTime + athlete3TotalTime + athlete4TotalTime) / 4;

outFile << athlete1Letter << " " << athlete1Time1 << " " << athlete1Time2 << " " << athlete1Time3 << " ";

outFile << athlete2Letter << " " << athlete2Time1 << " " << athlete2Time2 << " " << athlete2Time3 << " ";

outFile << athlete3Letter << " " << athlete3Time1 << " " << athlete3Time2 << " " << athlete3Time3 << " ";

outFile << athlete4Letter << " " << athlete4Time1 << " " << athlete4Time2 << " " << athlete4Time3 << " \n";

outFile << "Athlete " << athlete1Letter << " completed the race in a total of " << athlete1TotalTime << " minutes\n";

outFile << "Athlete " << athlete2Letter << " completed the race in a total of " << athlete2TotalTime << " minutes\n";

outFile << "Athlete " << athlete3Letter << " completed the race in a total of " << athlete3TotalTime << " minutes\n";

outFile << "Athlete " << athlete4Letter << " completed the race in a total of " << athlete4TotalTime << " minutes\n";

outFile << "The average total time for completing the race is " << avgTotalTime << " minutes\n";

inFile.close();

outFile.close();

system("pause");

return 0;

}