**CS173: Intermediate Computer Science**

**Reading 4**

Name: \_\_\_\_\_\_\_\_\_\_\_\_Daniel Lee\_\_\_\_\_\_\_\_\_\_\_\_\_

Read the assigned pages below from our course textbook. Submit your work by the assigned deadline on the Canvas course page or in class. Responses may be neatly handwritten or typed. **Put your name at the top!**

Readings: From the course textbook please read Chapter 4 – you may skip the section on GUIs.

You should come away with understanding:

* how input and output work in a C++ program
* how to open and read/write files in C++

**1) What type are cin and cout and where are they declared?**

The type of cin is istream and the type of cout is ostream.

They are both declared in the header file iostream.

**2) Explain the purpose of the “reading marker” and its role in parsing an input stream.**

The purpose of the “reading marker” is to keep track of the point in the input stream where the computer should continue reading, which plays a role of guiding the next character to be read while parsing an input stream.

**3) What role does the ignore function play in parsing input streams?**

The ignore function is used to skip (read and discard) characters in the input stream.

**4) When we normally read a string value using cin we get the group of characters separated by spaces. How do we read an entire line of text into a single string variable?**

We read an entire line of text into a single string variable by using getline() function.

**5) Write a C++ program that opens a file called temperatures.dat. The program should read six temperatures (floats) from the file and output the average temperature. Write or copy & paste your solution here.**

**#include <iostream>**

**#include <fstream>**

**using namespace std;**

**int main()**

**{**

**ifstream fileTemps;**

**float temperature, average;**

**float sum = 0.0;**

**float average;**

**int count = 0;**

**fileTemps.open("temperatures.dat");**

**for (int i = 0; i < 6; i++)**

**{**

**fileTemps >> temperature;**

**sum += temperature;**

**}**

**average = sum / 6;**

**cout << "The average temperature is: " << average << endl;**

**fileTemps.close();**

**return 0;**

**}**

**6) Briefly explain the design philosophies of functional decomposition and object-oriented programming (be specific about what each approach entails):**

Functional decomposition is a technique for developing software in which the problem is divided into more easily handled subproblems, the solutions of which create a solution to the overall problem. The design philosophy of functional decomposition is viewing the solution to a problem as a task to be accomplished and focusing on the sequence of operations that are required to complete the task.

Object-oriented programming is a programming technique for developing software in which the solution is expressed in terms of objects. Objects are self-contained entities composed of data and operations on that data. The design philosophy of object-oriented programming is solving a problem by identifying the components that make up a solution and identifying how those components interact with one another through operations on the data that they contain.