

PROG23672-Trafalgar-Winter2020-Assignment 1

Due date: Jan 31, 2020 11:59 pm

Submit to SLATE at DROPBOX

The men's long jump competition at the 2012 Summer Olympics in London, United Kingdom was held at the Olympic Stadium on 3–4 August. Performance data collected from this competition was recorded and named jump.txt which is attached with the assignment. This file contains the first names, last names, nationality and distances jumped by athletes in this competition.

In this assignment, you will develop a C++ solution to do the following: (1) read all data from jump.txt; (2) keeps the data into a vector of class objects; (3) prints out the names, nationalities and the distances of the athletes who exceeded a distance given by the user. The purpose of this assignment is to get the feel of object oriented programming using C++ and to construct an *Athlete* class which is derived from the *Person* class. Additionally, you will define and use a vector of *Athlete* objects as a collection data structure that keeps the competition records.

Description of the data file jump.txt file

The first line in the file contains titles and each other line in the file represents a data of an athlete so that white spaces separate the first name, last name, nationality, and the distance as shown below.

FirstName LastName Nationality Result

Aleksandr Menkov Russia 8.09

Aleksandr Petrov Russia 7.89

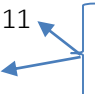
A sample of the output/input of the C++ solution including comments for clarification

The data is loaded successfully. ←(This line will be printed only if the data are loaded from the file successfully)

Please enter the distance threshold: 8.10 ←(This distance threshold will be entered by the user)

The athletes exceeded 8.10 m are:

Marquise Goodwin United States 8.11
Mauro Vinicius da Silva Brazil 8.11



(All Athletes jumped a distance that exceeds the threshold will be printed using the overloaded operator <<)

Requirements and resources

- The *Athlete* class must be derived from the *Person* class that models first name, last name, and nationality.
- Your solution must use dynamic memory allocation (allocates memory in the heap).
- The code must be organized so that header files contain class interfaces and CPP files contain implementations.
- Your solution must overload the operator<< to print the *Athlete* data including the jumped distance.
- Your solution must define and use a collection (a vector) of athletes in order to keep all data into the memory.
- Should you need more information about C++ and STL Library(vector class), please refer to lectures on SLATE and <http://www.cplusplus.com>
- Should you have questions please use SLATE email to send me them or book an appointment with me if needed

- Deliverables are a single zip file that includes all files of the visual studio solution or all C++ files (.cpp and .h files) if you are using Xcode

Evaluation Rubric

Learning objective	Level1	Level 2	Level 3	Max Mark
Validating the inputs	0-4 0-40% of the inputs are validated	4-8 40-80% of the inputs are validated	8-10 At least 80% of inputs are validated	10
Writing appropriate comments	0-4 0-40% of comments are appropriate	4-8 40-80% of comments are appropriate	8-10 80% -100% of comments are appropriate	10
producing a modular design	0-5 At least a class is developed + the main function	5-10 At least a class is developed + 2 to 3 meaningful functions	10-15 At least 2 classes and 3 meaningful functions are developed	15
Allocating dynamic memory in the heap	0-6 The project uses the automatic memory allocation in stack or a memory error occurred	6-12 The project partially use the dynamic memory allocation and the solution works fine	12-20 The project uses dynamic memory allocation and de-allocate them successfully	20
Loading the data from jump.txt into a vector of class objects	0-12 0-40% of the code is logically right. The data are loaded partially or a memory error occurred.	12-24 40-80% of the code is logically right and the program load data from the file successfully.	24-30 80% -100% of the code is logically right and the program read the file and print Athletes exceeded the threshold distance	30
Implementing inheritance and operator overloading	0-6 0-40% of the implementation of inheritance and operator<< overloading is correct	6-12 40-80% of the implementation of inheritance and operator<< overloading is correct	12-15 80% - 100% of the implementation of inheritance and operator<< overloading is correct	15