

Release necessary

CS22510- C++, C AND JAVA PROGRAMMING PARADIGMS

LANGUAGE CHOICE AND COMPARISONS OF ASSIGNMENT 1

for document title

Author:

Craig Heptinstall (*crh13*)

Version 1.2, Release

Department of Computer Science
Aberystwyth University
Aberystwyth
Ceredigion
SY23 3DB

April 12, 2013

Contents

1	Introduction	3
1.1	Purpose of this Document	3
2	Language Choices	3
3	Comparisons of Languages Chosen	3
4	Discussion of Alternative Options	4
5	References	5
6	Document History	5

1 Introduction

1.1 Purpose of this Document

This report provides my solution to assignment two of CS22510 (C, C++ and Java) regarding the three programs I created in assignment one. I will attempt to answer all the required specifics of the brief [1] given to me to the best of my ability. I will reference any pages or useful guides I came across in the references section of this report. I would just like to note that the 2000 work word limit has been kept here, and to ensure this document in within it, I used an on-line Latex word count program [2].

2 Language Choices

I will now briefly discuss the language choices I made for each of the programs created in the previous assignment of this module concerning the runners and riders event creator, checkpoint manager and event manager systems. Starting with the first of these, the event creator program required me to create a system that could allow users to input the event name, entrants and tracks. Because this program required only reading input from the user and then writing this to the specified file, I felt that C++ would be the ideal language to be used.

My reasoning for this was firstly the 'cin' function available within the language. The cin was a great tool as it allowed for a wide range of inputs under one function unlike having to identify the specific type that was being read in, for instance a String, Int or any other type. This meant I could easily read in a range of inputs for the required files that would be written. This was similar for the output, as 'cout' easily allowed output of text to the console. To summarise this, the bitwise operators '>>' and '<<' was at the heart of my choice here because of the ease of input and output of a program where this was the key functionality.

Looking at the next section of the runners and riders that also needed implementing in one of the three languages was the checkpoint manager program. For this, I chose to implement it with the Java programming language. When I was handed this assignment and found out that a graphical interface based program would be a required part, I instantly decided that Java would be the best option. My reasons for this language mainly lied upon the way I could easily create the GUI's and transfer data about from user input to the forms. Although the Gui was not the main part of this program, I still think that by using Java it made it much easier to create the interface quickly allowing me to concentrate on the use of reading and writing files to the users desire. I would also like to add that because Java is a portable language, and the system here would be intended for being carried around a track when updating checkpoints, it was an ideal solution if it was on a range of systems.

The final program in the first assignment was the one I used the C language to create. This was the program for the general manager of race events allowing several options for the user, all in command line. Because C is a powerful language, and this program was to be a command line system, this is the start of my reason to choose it for such a program. Firstly, the simplicity of moving around and checking data was ideal for the wide range of functions that are implemented. For instance, using Structures in C was a easy way to store sets of data for an entrant. The main reason I chose C was the strictness that the language brings. It meant that if any data did not relate to the header file available, or methods were implemented that should be, the program would report an error. This was very important to me because this part of assignment one was the most important because of the wide variety of options in the system.

3 Comparisons of Languages Chosen

For the next section of the report, I have been asked to compare and contrast these different languages taking into account my choices for which languages I chose to implement the different sections of the system with. Therefore, I will be firstly addressing the use of input and output for each of the languages. Starting with C, this was the language I found that although could read in the same types as C++ and Java, it required more code and work. This was because I had to specify what type was being read in for instance a float which was %f rather than a %d for integers. Java works on a similar principle although can handle floats and integers being read in using the same method. Meanwhile C++ can handle a wider range of input types under the same cin method . I found that although this meant it was easier to code, C was the best performing here because of the stricter method. It meant that any wrong types could not be inserted into the wrong format variables.

As for the outputting methods of each of the languages, this was the similar story to the inputs of each. When I was outputting in Java, this was the simplest of the three, by using using the 'println' statement. In C++, similar to Java in the sense of outputting any types into the console in one line, although I did not know how to

output multiple variables at once. After looking this up I found using the binary operators '<<' was required after each variable or text. Hence Java being the best in comparison for outputting onto console.

The next set of comparisons I want to look at is the way I stored data with each of the three languages. For instance, Java offers an alternative to the other two, by using Objects rather than structures [3]. In C and C++ parts of the first assignment I found using structures somewhat harder to use at first, being because that structures cannot have methods inside each structure while Java could. Because of this, it emphasised my reason for the Java choice for the checkpoint manager. This leads me onto the next major difference concerning the storage of multiple entrants for instance in an array. While Java allowed me to create an ArrayList, meaning I could always add to it easily, C and C++ required the use of malloc and calloc. This made Java a lot easier to use when juggling around arrays of entrants, while in C and C++ for simplicity I used set sized arrays and made them to a size I would find suitable for any data inputted for an event. I should also mention that pointers are an item of the C and C++ languages that Java does not have an equivalent to. This is because Java passes data by value rather than having the option to pass by reference unlike the other two. This meant easier and wider options of moving data around my programs for the event creation and event managers.

The final two compare and contrasts concern the strictness of the languages and also the graphical capabilities for each language. Especially in C++, I found that this language uses very strict type checking. For instance, a prototype must be known for each function that is called, and the call must match that prototype. In Java the language is much more flexible, and C similarly that does not have such checking but does allow header files which ensure types are implemented and signified.

Looking at the graphical user interface implementation possibilities of the three languages, because I have had a good deal of experience in Java, this swung my decision to create the checkpoint manager in it to save time during the project. I must also mention that from looking at tutorials and discussions concerning GUI's in C and C++ [4]. In my experience of trying the three, Java was the easiest because of its built in libraries for the creation of GUI's meanwhile C and C++ required additional libraries such as QT. The portability of Java is usually considered much more portable than both C and C++ because of the built libraries. Between C and C++, C is more portable than its counterpart because of the need for less libraries. [5]

As a last point, regarding the code layout and comments of each program, Java has the Java Doc which provides in line commenting of methods and variables automatically whilst C and C++ does not provide this without additional support. Java files are all usually placed in packages as I used in my program, yet the other two have their files usually all together and to call another method the program must import the required classes. Although in Java these classes represent as objects. Generally, the speed is also faster from looking at reviews and general information pages [6] than Java at compile/ run time when it comes to the two other languages.

4 Discussion of Alternative Options

Now I have completed both the first section regarding the choices I made considering each of the three languages the entire system was created in, and compared each of these in terms of implementing code, I will now consider any possible changes I would make if I were to do a similar project again. For instance, choosing to use different languages for the programs in a different order to what I chose this time around.

The first change I would make looking at the event manager program. In the previous section, I said C++ was more strict than the other two languages, therefore I think if the manager program of the event was written in this language rather than C would mean it would be less error prone, and incorrect data would produce errors. Using C++ for the event manager would be ideal because this program was the one that handled more data and options to the users. order to what I chose this time around.

The next change I would have liked to implement if I did this project a second time around would be changing the program used to construct the GUI with for the checkpoint manager. Rather than using Java, although easier with my current knowledge it would be to construct, I think after learning C graphical programming I would be able to construct similar if not better GUI's. Firstly I noted previously that there are a range of libraries available to construct graphical programs with C available. This would give me more of an option in my choice of elements contained in the GUI. I also noted that speed was a difference between C and Java with C being faster. This would mean that the graphical programs would load faster for users which would provide especially useful for people quickly attempting to register a runner/ rider on the fly. The only issue I would have to overcome with this change would be that It could prove more of a problem concerning portability using C over Java for such a program with the extra required libraries.

Finally, With the first program from assignment one, which would be the last remaining out of my choices for a language, Java would be the one to program this in. Apart from choosing this as the last language and program remaining, there are still reasons to choose this language appropriate for this type of system. For instance, rather unlike in C where I had to have structures string data in the same C file, Java would allow me to easily separate data being inputted into separate classes essentially turning them into objects. This would organise the data better and keep objects separated clearer.

Word Count: 1966

5 References

References

- [1] CS22510 Assignment 2 Brief. Neal Snooke, Fred Labrosse. 2012-2013.
- [2] TexCount Web Service. V2.3. <http://app.uio.no/ifi/texcount/contact.html>.
- [3] Differences in Java and C/C++, including data storage. <http://www.programmersheaven.com/2/FAQ-JAVA-Differences-Between-JAVA-And-C-CPP>. 2010.
- [4] Stack Over Flow. <http://goo.gl/eY4uq>.
- [5] DI Management Site. Why C is better than C++. <http://www.di-mgt.com.au/cprog.html>. 2010
- [6] Prof. Cliff Click Blog. <http://www.azulsystems.com/blog/cliff/2009-09-06-java-vs-c-performanceagain>. 2009

6 Document History

Version	Date	Changes made to Document	Changed by
1.0	April 7, 2013	Creation of first draft of document	Crh13
1.1	April 11, 2013	Amendments and spell checking of document	Crh13
1.2	April 12, 2013	Release	Crh13