Programming Techniques

Week 1a: Introduction

01/2024

Description

- The course provides an introduction to software design, use of a variety of data structures, data abstraction, and recursion.
- Program correctness, verification, and testing is also introduced.
- Students will be asked to write a substantial computer program during the term (in C++), providing a user's manual and external design documentation.

Topics

- Topic #1: Introduction and Review
- Topic #2: Pointers and Dynamic Memory
- Topic #3: Singly linked list
- Topic #4: Other linked lists
- Topic #5: Stack and Queues
- Prepare for the Midterm & Midterm exam
- Topic #7: Recursion
- Topic #8: Recursion (cont.)
- Topic #9 Arrays with Structured Elements
- Topic #10: Binary Files
- Topic #11: Prepare for the Final Exam

Exams and Marking

- Assignments + project: 30% (=20%+10%)
- Mid-term exam : 20%
- Final exam: 50% (=40% paper+10% lab)
- Others (active in the class, discussion board etc.) ➔ bonus scores

Any plagiarism found → 0 for the whole course

Books and references

- Books
  - C++ Primer Plus, Stephen Prata, SAMS.
- Other references:
  - The C++ Programming Language, Bjarne Stroustroup.
  - Nhập môn lập trình, Khoa CNTT, ĐHKHTN
  - Kỹ thuật lập trình, Khoa CNTT, ĐHKHTN

The project

- In this course, project is bigger and more complicated than the Introduction to Programming
- The intent is to incorporate all that you learned in the introduction to programming to solve a “real world” programming problem
- And, apply some of what we will learn this term in the assignment.
- It is large, so begin on it as soon as possible!

The project

- One of the goals is to learn how to create a user friendly environment
- This means you should assume that the user doesn't know anything about computer programming
- This means you will need to carefully prepare prompts, echo all input, provide labels for your output, and...

The project

- Error check (i.e., the user may enter invalid data).
- The types of things to check for include:
  - User typing in too many characters.
    - make sure to throw these away using cin.ignore
  - User types in an incorrect option
    - you prompt for options 1-5 and they enter 99
  - User types in lower versus upper case
    - you should accept either! (Y, y, N, n, No, NO, YES, yes,...are all valid confirmations!)

The project

- In your project you will need to use:
  - structures and array of structures
  - pass all objects of a structure by reference --- NEVER by value!
  - external data files (fstream)
  - your main program should be very small

The project

- In your project keep in mind:
  - Use call by reference instead of call by value whenever possible to improve efficiency.
  - Use iostream (and fstream) libraries. Do not use stdio.h for your I/O
  - Display a menu of items the user can select from. Remember to allow the user to quit!

Review of the last course

- Let's list the areas that you would like us to review this week:
  - pass by reference vs pass by value?
  - defining arrays of characters?
  - reading strings using 3 argument cin.get?
  - structures? arrays of structures?
  - passing structures by ref vs by value?
  - reading/writing external data files?
  - others?

Review

- Why use call by reference?
  - supply a value back to the calling routine
  - more effective use of memory
- Why use call by value?
  - only when you need a spare and duplicate copy of the data or if passing fundamental data types (like an int, short, char)
- Why use constant references?

```
void print(const float& data);
```

Review

- How is an array passed to a function?
  - what does the function call look like?
  - what does the prototype look like?
  - is there any way to pass an array to a function by value? vs. by reference?
  - this term it is important to realize that the name of an array is a constant address of the first element in the array. It is that which is passed (by value)!!!!

Review

- Reading in arrays of characters:
  - what is the advantage/disadvantage of:

```
char s[20];
cin >> s;
```

  - what is the advantage/disadvantage of:

```
cin.get(s, 20, '\n');
```

  - what does this do:

```
while (cin.get()!='\n');
```

or,

```
cin.ignore(100,'\n');
```

Review

- Reading in arrays of characters:
  - after using cin >> any_variable;

    what is left in the input buffer?

    what will it do to a subsequent call to:

    ```
    cin.get(s, 20, '\n');
    ```
  - Remember, cin.get does not skip leading whitespace (nor does cin.getline).
  - cin.getline should not be used this term, as some compilers will hang if the user types in more than the specified # of characters!

Review

- What is the purpose of a function declaration (i.e., prototype)
  - to allow a function to be called even if it is defined (i.e., implemented) later or in some other file.
- What about defining arrays,
  - can the size be variable? (no!)
  - remember to allow 1 character in a “string” for the ‘\0’ (terminating null)