MURDOCH UNIVERSITY

ICT106 Fundamentals of Computer Systems
Semester 2, 2006
C Programming Project

Project Abstract (5%) Due date: Monday 28\textsuperscript{th} August 4PM (Teaching Week 5)

Final Report (15%) Due date: Monday 16\textsuperscript{th} October 4PM (Teaching Week 11)

Submission

Internal Students (Murdoch): Post your submission in the ICT106 Assignment box located outside the School of IT office at ECL 3.037.

Internal Students (Rockingham): Post your submission in the Assignment box at Level 2, ACL Building.

External Students: Post to External Studies Office by the due date as shown above.

Late penalty: 10\% per day penalty for delayed submissions unless prior extension of deadline is obtained from the unit coordinator. Note that extension is only granted for situations due to unforeseen circumstances beyond one’s control. Examples are sudden illness, death of family member, accidents or natural disaster. Too many assignments, tests, heavy workload, or lost of data due to computer failure are not considered as unforeseen circumstances as you should have been told in advance all the assignment due dates.

Back-up Copy: You must always keep a copy of your work. Your submission must include a completed (and signed) assignment cover sheet. An electronic copy of the assignment cover sheet is available in the Assignments folder at the unit resource site.

Requirements

In this project, you are required to design and develop a C program covering as much as possible all the operations and programming constructs taught in the course. In addition, the program should serve specific application objectives. The program should be based on ANSI C to ensure its portability. Note that it is not essential for your program to include extensive Graphic User Interface (GUI).

This is a list of suggestions for such programs:

- **Edutainment Games** for school children teaching simple arithmetic or geometry.
- **Guessing Game** – to guess a mystery number within a specified number of trials.
- **Encryption/Decryption Game** – given an encrypted statement, can the user find out the message?
- **Travel Calculator program** – Record the distance travelled and amount of fuel used then calculate the average consumption. Make some suggestions to the user on how to save fuel.
- **Exercise and/or Calories Counting** – Relate the type of exercise, duration, level to the amount of calories burnt and compare to the calories intake. Calculate the weight gain/lost.
**Project Abstract 5% (Due date Week 5, Monday 28th August)**

In the abstract, you need to provide a description of the program to be developed. This should include:

- Objective of the program (What do you try to develop?)
- Significance (Why is it important to develop it? Any commercial value?)
- Initial design (How will you do it? Give a block diagram of the system.)
- Identification of Input, Output, Algorithm… (An overview of the program but with more details on the input and output.)
- Theory of the Algorithm (A description of the processing principle)
- References (Where did you get your information? Make sure that you do not take other person’s work as yours!)

Your tutor will assess your abstract and provide you with feedback on the abstract. You will be informed that whether you are on track, too much, too complex or too little. If it is too much, the project will unlikely to be completed and you have to drop some of the functions. Too complex means that the algorithm might be too difficult for your level. You might have to opt for a simpler and manageable solution. Too little means that your proposal is too trivial for the duration and the expected technical skills. You’ll be advised to increase the complexity or functionality of the program.

**Final Report of the Project 15% (Due date Week 11, Monday 16th October)**

After the submission of the Abstract and including the week break, you’d effectively have 7 weeks to work on the project. You should continue with the development based on the initial design and adopt a “modular” approach to carry out the development. Your project should also base on programs or exercises from the practical sessions. However, you should consider carefully how the functions or modules are linked to one another. Make sure that you’ll manage the time effectively and do not expect to complete the program in just a few days before the deadline. The following sections will give further information concerning the project.

**Marking Schemes:**

- Documentation (includes: report, pseudo-code, assumptions made and limitation)
- Program functionality and design (includes: requirement functionality and error handling)
- Testing (give a plan on how to test your system in order to make sure it works according to the specificiations. Data sample should be provided.)

**Essential Requirements:** The report should be comprehensive and the program should be well-structured. The program should have a reasonable set of functions in addition to the main function. It should use meaningful variable names, comments and good layout. Your submission is required to demonstrate good documentation and programming style.
NOTE:
You should develop and test the program in stages.

i. Write and test a simple program to read in the data input (such as options) and display the input and results correctly.

ii. Keep a copy of this working program.

iii. Edit a copy to incorporate one component (eg, functions for one option) and test that component. Make sure that it works and save a copy of the working program.

iv. If a component gives errors at any stage, you should leave that for a while to think about it. Go back to an earlier version of the program and try incorporating and testing a different component.

v. You must always keep a working version of the program with its test results so that you have something to submit.

vi. For components that do not perform as expected, submit a listing annotated with your ideas on the cause of the problem and possible solutions.

vii. You will be given marks for the work that you have done. In other words, even the program is not fully functional, your work on the design and testing will receive some marks.

viii. You may not receive full mark just because your program fulfils the requirements. You may lose marks if your work do not fulfil all the essential requirements (such as documentation and programming style.)

ix. **Non submission means no mark.**

What to submit
Your assignment submission should include:

- A **hard copy (printed version) of your report and the program** including internal documentation (see below) must be submitted.
- A **hardcopy of test runs** containing both input data and output results must be submitted.
- **External documentation** (see below) must be submitted
- A **disk or CD** containing your source code and an executable version of your programs must be submitted unless other arrangement with the tutor.
- Internal students should **test the compilation and execution** of their program on the School of IT lab machines.

For internal documentation (ie in the source code) we require:

- a beginning comment clearly stating title, author, date, file name, purpose and any assumptions or conditions on the form of input and expected output;
- other comments giving useful low-level documentation and describing the each component;
- well-formatted readable code with meaningful identifier names and blank lines between functions.
**Required External Documentation**

1. **Title**: a paragraph clearly stating title, author, date, file names, and one-line statement of purpose.
2. **Requirements/Specification**: a paragraph giving a brief account of what the program is supposed to do. State any assumptions or conditions on the form of input and expected output.
3. **User Guide**: instructions on how to compile, run and use the program.
4. **Structure/Design**: Outline the design of your program. Give a written description, use diagrams and pseudocode.
5. **Listings**: attach source code listings, i.e., a print out (hardcopy) of program source code
6. **Testing**: describe your testing strategy (the more systematic, the better) and any errors noticed. Give hardcopy of results of testing.
7. **Limitations**: Describe program shortfalls (if any), eg, the features asked for but not implemented.

**Remember to complete and sign the Assignment Cover Sheet and submit it with your work.**

Lance Fung, ICT106 Coordinator, 12th July, 2006