ICS4U – Assignment – Final Project

You will plan, design and code a software program of your choice. The design and development of your program will be groups of up to 2 students. You can ask for help, but your ideas AND code must be your own!

Your program when complete must have the following:

* Repetition of code through loops (for and while)
* Decisions (if elif else structure)
* ‘thinking’ or AI logic component
  + i.e. computer player in card game or equivalent
* Intuitive/easy interface for user(s)
* Include GUI (at least some)
* Try and catch
* Methods
* Utilize arrays and/or linked lists to store and manipulate data
* Utilize Classes – object orient programming
* Write to and read from text file(s).

Your program, if applicable, must have an “open/exposed” version. This version should print key information on the screen for program analysis (for example reveal the hidden number that the user is trying to guess).

Almost any idea you come up with is acceptable as long as it fulfills the requirements above and is complex enough; see suggestions below.

Submissions – You will be required to submit your project in 5 phases. All submissions will be submitted to 4students/IN/…../

Phase 1 – Rough Proposal Due Tue Dec 19 2017

Research and develop problem / program. You will research and propose your project. The complexity of the project will define the maximum attainable mark for the project. Submission should be a ‘rough’ proposal. **If creating a game please include a complete set of rules (point form preferred). The rules will be modified to adjust the complexity of the program to an appropriate level.**

Phase 2 – Final Proposal Due Thurs Dec 21 2017

You will provide the scope of the program which will include but not limited to:

* main program written (pseudo code) with methods
* identifying inputs
* identifying all process(es) and methods() – description or code
* identify who is doing what
* Display appearance (what the user will see) – detailed description
* “Rules” if applicable – a complete description of the rules will be provided here (usually associated with games).

Phase 3 – Daily activity log due daily

Implementation through code. You will turn your ideas into code. During this phase a daily log will be kept that tracks your progress. Your daily log will consist of two sentences due at the end of the period. One sentence to summarize what was done that day and one sentence to summarize the objective for the next day and one sentence to identify problems you are having/solving. You will submit a text file daily using the format “LastName\_FirstName\_yyyy-mm-dd.txt”. An example is given:

2017-06-01

Done – Menu created, user input read in, verified and stored into variable, user entering letters debugged, list function started.

Next – finish list function, create add function, create process for calculation of other values

Problems – trying to work out how to use indexes to access the data from array, figuring out how to use JTable.

Phase 4 – Beta version Due Thurs Jan 11 2018

Beta version submission. Your program must run and somewhat resemble the final program. If necessary, certain components may be implemented in a “rudimentary” form (for example if a method is still ‘work in progress’ you may have it return a default value to allow the rest of the program to run).

Phase 5 – Final Submission Due Thurs Jan 18 2018

Proposal Marking Rubric

|  |  |
| --- | --- |
| Daily Log – complete and kept up to date; easily understood, includes problems and solutions | /5 C |
| Display – easily understood, complete/fully described – good layout | /5 C |
| Methods – inputs clear, specific, identified; process - all processes and functions identified and clearly described; outputs - clear, specific, all required outputs included | /10 T |
| Data – all data that is required is identified including type | /5 T |
| Discussion(s) – student demonstrates understanding of programming concepts: class, object, constructors, arrays, index, conversion/casting, methods, parameter/arguments, try/catch, decisions, (random API class) | /10 K |

Program Marking Rubric

|  |  |
| --- | --- |
| Proper header included at the top of all programs. | /1 |
| Variables named appropriately for easy decryption. Spacing used appropriately (improves readability); not too much/little. | /2 |
| User interface easy to understand, include prompts, clear instructions, navigation is easy. Code is efficient. Buttons intuitive. | /6 |
| Test cases proper input, improper input, program data is accurate, game follows rules, computer play indicates knowledge of game strategy/AI thinking, | /8 |
| Total | /17 A |