TRINITYHOUSE HIGH SCHOOL

INFORMATION TECHNOLOGY

Grade 12 PERFORMANCE ASSESSMENT TASK

Programming Project

You are required to use the software development cycle to produce a single programming application that is based on a theme of your choice.

The program/application needs to meet the following criteria:

* **User friendly** –
* **Interface** - the GUI interface must be user friendly, easy to use and task appropriate.
* **Data Flow / program operation** – sequence of steps required to use the program and complete a task are clear, easy to follow and logical.
* **Storage / Data persistence –** data must be stored and retrieved from session to session (conventional files OR a database OR both). Use of a database and SQL is preferable. The storage must be appropriate to the program (e.g. a game should have the ability to save & load a game & have high scores, etc)
* **Separation of interface & engine** – the ‘working code’ **must not be embedded in the interface** (i.e. it must be in separate classes). Communication between the interface & the working code is in the form of parameters & typed methods. A limited amount of code in the interface is acceptable only if suitably justified in the planning.
* **Good internal data structures** – There has to be some form of internal representation of data (i.e. classes / records / arrays). Data structures must be logical and task appropriate. Classes must be used.

Design & development of your application:

* according to the Software Development Life Cycle (Learning Unit 6 of exploring IT: Java Programmig Grade 12 by Delia Kench
* hand in each deadline as per the agreed upon project schedule
* **It is vital that there is evidence of planning in your project.**
* Ensure a simple program that works 100% and satisfies all the criteria.

Final due date: **Tues 24 July 2018**.

* Late submissions will penalised with a 5% deduction for each day late.
* Hand in the following:
* Project Specification Document
* Design Document
* Technical Document
* A CD/flash stick containing the program code, compiled program and all documentation, as well as comprehensive and valid data files.
* Referencing – may use other sources (books, Internet, friends, teachers) for up to 20% of coding. Reference all sources using the Harvard Standard in the appropriate document
* Authentic work - sign a ***Declaration of Authenticity document***.

**Details of documentation requirements:**

1. Project Specification Document
   1. A title page: Include the project name, ‘Project Specification Document’ and your name.
   2. A table of contents.
   3. A summary of the features of the project - an explanation of your success criteria and what the program actually does. These features should be very specific..
   4. Project Goals and Success Criteria – a description of the goals that are to be achieved by the project and a list of criteria against which the project can be measured to determine the success of the project.
   5. A basic description of the user interface, help and storage needs for the program needs to be mentioned.
   6. Security requirements if there are users and administrators for the program.
   7. Hardware, Software and Installation Requirements.
2. Design Document
   1. A title page: Include the project name, ‘Design Document’ and your name.
   2. A table of contents.
   3. Summary – this section must describe the all the parts of the problem that are addressed, and should include a description of the detailed architecture of the project.
   4. Presentation Layer – this section should describe each screen and report layout of the project. Graphical User Interface Design showing navigation of screens. These can be done in Java.
   5. Business Layer – a graphical representation of the flow of information through the system (using Visio), as well as a detailed tabular description of each class in the project.
   6. Database Layer – design, table relationships, fields in each table etc. You must also include a justification of why you have used that particular storage.
   7. Design of on-line Help – what sort of help is going to be included and how is the user going to access it.
   8. Test plans and test data (generated and/or selected using standard, extreme and abnormal – depending on which is most appropriate)
   9. Evidence of testing.
3. Technical Document
   1. A title page: Include the project name, ‘Design Document’, your name.
   2. A table of contents.
   3. References to any code that you used that was not your own – (max 20% of code)
   4. Critical Algorithms/flowcharts i.e. algorithms that have a significant effect on the performance of the program – not allowed to be generated from the code.
   5. Sophisticated techniques used - list and explain how they have been used. A sophisticated technique is anything that falls outside of what has been taught in class.
   6. A hard copy of the program source code.

**Details of program requirements:**

* 1. Project should not depend on specialised equipment. Your application must run on computers available at school.
  2. Comments - sensible and consistent i.e. complicated sections of code need explanation and each procedure or function should have a brief description directly following or preceding the procedure/function name. Eye-catching use of asterisks or dashes enhances the readability of the code. It is wise to build comments into the program during development, and not leave it until the end.
  3. Structure: Structure is the division of a program into modules/classes, with parameters, that each perform a single task. Also the use of control structures such as loops, if and case statements. The program must be well structured so that its logic is easy to follow. Good structure comes from good planning.
  4. User-friendliness: The program should be as easy and pleasant to use as possible. Some of the following ideas can be used:
* A built-in Help system
* Use of Hints for input
  1. Output: Keep the output simple but self-explanatory. Cluttered screens with many graphics and/or too much text are off-putting. The output of data should use appropriate display methods, and should be clear and logically arranged.
  2. Other important points:
* There must be no duplication of code unless this is unavoidable.
* The most efficient solution should be used.
* The program could be enhanced by incorporating graphics.
* On-line Help that is provided should be contextual and general.
* Screens should be well designed and clear. The design should be consistent.
* The method of input of data should ensure that as much automatic input as possible is used, and that this input is validated where possible.
* Include data files with a reasonable amount of data that is all valid.
* Include comments that explain the use of each class

# Programming Project – Project Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | | **Due Date** | **Description** | **Marks** |
| **What must be submitted by the candidate:** | Project Specifications |  | List (and describe) the functions that your program needs to achieve in order to be a 'success'. | 14 |
| Design Document |  | Design the user interface, sequencing (data flow), class and persistent storage of the program in detail. | 31 |
| Coding & Technical Document |  | Write the program following good programming techniques and document it by printing the code & explaining critical algorithms. | 50 |
| Testing Document |  | Document what is to be tested, the test data used and the results of the testing. | 5 |