

# Assessment Task Notification

## In Class Task

# Year 12 Software Design and Development



### Context

This is the next stage in the SDLC employed in this course. Students are continuing to develop the personal project as an opportunity to apply the theory covered in this course to a practical project.

<b>Task:</b>	Task Number 2 Design Presentation		
<b>Date:</b>	Term 1, Week 8, Thursday		
<b>Weight:</b>	25%	<b>Marked Out of</b>	60

### Outcomes

- H2.1 explains the implications of the development of different languages
- H4.1 identifies needs to which software solutions are appropriate
- H4.2 applies appropriate development methods to solve software problems
- H5.2 creates and justifies the need for the various types of documentation required for a software solution
- H6.2 communicates the processes involved in a software solution to an inexperienced user
- H6.4 develops and evaluates effective user interfaces, in consultation with appropriate people

### Requirements of the Task

This is a presentation to the class on the progress made to date on your Major Software project.

Your project is to create a software solution to an identified problem of your choosing. In developing the software, you are to be applying the Structured Approach to software development to all stages.

- You are free to choose the language/development platform for your solution. Your software solution will need to present some form of user interface.
- Your software solution may be presented as a prototype with some features only represented as mock ups within the supporting documentation or website or 'dead links' within the solution itself. The areas of your software solution that you do choose to present as functional must be completed to a high and consistent standard.
- Your code will not be marked directly, however you will need to be able to prove maintainability (intrinsic documentation features) of your code and also acknowledge any sources of code snippets, tutorials or libraries as outlined in the supporting documentation. Maintainable code will also result in easier testing and improved solution quality.
- Be free of functionality, spelling, grammar, punctuation, runtime and logical errors
- Your software solution should be free of inappropriate, violent or offensive content and should not represent the class, this school or the department in a negative way.
- You will need to be willing and able to demonstrate functionality and aspects of your interface to the class during and after development and your documentation and solution should align.

Your software solution must contain:

**Processing/manipulation of some form of input provided by the user** and/or external files Sophisticated and appropriate use of the three control structures:

- Sequence
- Selection
- Repetition

Sophisticated and appropriate use of a selection of **data structures** (or equivalent) aligned with your data dictionary:

- Arrays, multi-dimensional arrays
- Records, arrays of records
- Files

Sophisticated and appropriate use of a selection of the **standard algorithms** from the Preliminary and HSC courses. Remember that you need to be able to (\*individually) represent three of these in the supporting documentation in pseudocode and flowchart form:

- String processing
- Searching
- Sorting or min/max
- Printing/accessing/processing of arrays (or equivalent)
- Opening/loading/writing to/closing files

**Maintainability** features for snippets demonstrated in the supporting documentation:

- Intrinsic documentation – appropriate variable and function names, commenting, indentation, order of files, libraries and modules.
- Measures to prevent/limit user, file or internally produced errors

#### **Ergonomic considerations**

- A quality, consistent and functional graphical user interface with appropriate placement of screen elements.

**Consistency** of language used, elements sizing, interactions, commands, layout and formatting.

- Instruction, labels and navigational cues to the user where appropriate and expected
- Acceptable response times with feedback to the user
- Suitable for target audience with appropriate rating to indicate appropriate users
- Inclusivity issues features for the identified target audience
- Copyright/about/version information available in solution
- Security provisions/disclosures if solution utilises private or personal user data

#### **Supporting Documentation**

You will need to ensure that your project is effectively managed (these will be required within the assessment task and at point-in-time progress checks throughout).

You will need to select a range of these, dependent upon your project.

- manuals, incorporating screenshots, table of contents, index and footnotes
- algorithms (flowcharts and/or pseudocode)
- system flowcharts (using specialist software such as Visio or SmartDraw)
- structure charts
- data flow diagrams (using specialist software such as Visio or SmartDraw)
- context diagrams (using specialist software such as Visio or SmartDraw)
- storyboards (using presentation software such as PowerPoint)
- data dictionary (using spreadsheet software or a table in a word processed document)
- IPO diagrams (using spreadsheet software or a table in a word processed document)

Updates to Gantt charts and Logbooks are also a mandatory requirement.

### **Submission Instructions**

This task will be in the form of a presentation and demonstration on the progress on the task to date.

Students are encouraged to provide evidence of the application of the Structured Approach and the adherence to the original task expectations. This should include the use of a data projector and the ability to demonstrate and outline the work involved in meeting the task requirements.

### **Assessment Criteria**

You will be most broadly assessed on the work completed and demonstrated but you will also have a proportion of the marks awarded for the ability to present the information in a succinct and clear manner.

### **Feedback**

Students will be provided written feedback on all areas of the task.

Individual verbal feedback will also be provided to students on the direction and execution of all elements of the solution presented.

### **Illness Misadventure**

If you are absent for any reason on the day the task is due, you or your parent/guardian, must notify the course Head Teacher and the Class teacher.

You are to arrange with the Head Teacher a new date to complete this (or an alternate) task.

Within two college days of the missed task, you must submit a completed [illness/misadventure form](#) with supporting documentation to justify your absence.

## Marking Guidelines

Criteria	Marks
<b>Solution</b>	
Relative completion of the coded solution	5
Relationship of the coded solution to algorithm	10
Appropriate selection and application of data types and data structures	3
Evidence of the inclusion of a standard algorithm	3
Evidence of a modular approach	3
Error minimisation	3
Interface design standards	10
Manuals and help files – included as required	5
System documentation – updated as required	3
Project management tools – updated as required	5
<b>Presentation</b>	
Clarity of presentation	10