Software Dev – plan 2016

*(image below adapted from Chris Bucknell)*

SAC 1 U3O1

10% of study score

Folio of code modules

SAC 2

10% of study score

Written/visual report

SAT U3O2, U4O1

30% of study score

Written/visual report

U3 Area of Study 1

U4 Area of Study 2

2

U3 Area of Study 2

U4 Area of Study 1

Software: Prog language

Tool: project plan

Software Prog language with data storage

Tool: document project plan

Assessment

Software

SAC 1

Coding modules (ATs), demonstrating application of the skills & KK.

Will provide pseudocode, or IPOs or layout designs (AT6) with a mini scenario.

SAT part 1…

SRS

Alternative design ideas

Represent the preferred design

Project plan for Dev

SAT part 2…

**SAT requirements:**

* students individually identifying a need or opportunity that can be fulfilled through the use of purposed-designed software
Show students the assessment criteria, examples of needs, and guide students on their choices.
* Students should identify a real-world problem or opportunity – not a problem determined by the teacher. This could involve students engaging with an individual business, a sports club or a community group to identify a problem or need.
*(The notion that current products, processes, systems or services are either unsuitable for our needs or can be improved)*
* Student-determined problems/needs have the potential to engage and sustain student interest over an extended period of time.

Options could include: an app, web-based service, XML stream use, game with data store…

Places to look for ideas: your workplace, sports club, church, social issue, news, local organisation or business, …

**The SRS should include:**

1. the purpose and audience of the SRS
2. user characteristics (general characteristics of eventual users)
3. environment characteristics (technical description of environment within which the solution will operate)
4. the scope of the solution
5. functional requirements
6. non-functional requirements
7. constraints
8. context diagrams
9. data flow diagrams
10. use case diagrams.

*Taken from* [*http://www.digipubs.vic.edu.au/vcaa/vce-computing/vce-computing-units-3-and-4-software-dev-unit-3-intro#study\_1\_programming\_practice*](http://www.digipubs.vic.edu.au/vcaa/vce-computing/vce-computing-units-3-and-4-software-dev-unit-3-intro#study_1_programming_practice)

**Ideation – Idea generation**

Teachers should engage students in specific ideation techniques to help them generate ideas – this is an essential ingredient in creative thinking. Examples of ideation techniques include:

Brainstorming

Forced analogies (comparing an existing solution with something that has little or no connection with the solution)

Attribute lists (decomposing a solution and then recombining or improving individual elements to create a new form)

Active searches (targeted searching for a specific solution; this contrasts with passive searching that involves searching for inspiration but not for a specific solution).



**Exam Prep**

**AOS 3**

**AOS 3**

**AOS 2**

**AOS 1**

**Software Dev – weekly outline 2016**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Key knwldg | Key skills | In class (teaching and learning tasks) | At home |
| TERM 11 | AOS 1 |  | Before launching into the course…Read **Careers in Code magazine / website**. Choose two people or careers that speak to / inspire you. Share with the class why.  | Screenshot one of your inspirational articles. Share it on our SBox page wiki. |
|  | KK1, 2, 3, 9 | Variables, arrays | Data types, Internal documentation (commenting)Arrays - <https://www.codecademy.com/learn/php> Linear search <https://www.youtube.com/watch?v=JQhciTuD3E8>   | Practice arrays, numeric manipulation & linear search using a loop. |
| 2 | KK 6, 8, 12, 7 | Prog structures, functions | [What is an algorithm – and why should I care?](https://www.khanacademy.org/computing/computer-science/algorithms/intro-to-algorithms/v/what-are-algorithms) **AT1** = arrays + manipulation of text & numeric data. Linear search Programming structures - selection   **AT2** = simple selection statement and CASE/switch Programming structures – iterationTrace tables  | Practice CASE statements in PHP using [codeacademy](https://www.codecademy.com/learn/php)Practice iteration (loops) using [codeacademy](https://www.codecademy.com/learn/php) and write a trace table to test them. |
| 3 | KK 8, 7KK4 | Search XmlInput/Output  |  **AT3** = iteration x 2 examples, plus trace tables for each Binary search* <https://www.topcoder.com/community/data-science/data-science-tutorials/binary-search/>
* <https://www.youtube.com/watch?v=JQhciTuD3E8>
* Khan academy [w exercise](https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search)
* <http://infopotato.com/blog/index/binary_search>

*time your search in php* [*http://stackoverflow.com/questions/6245971/accurate-way-to-measure-execution-times-of-php-scripts*](http://stackoverflow.com/questions/6245971/accurate-way-to-measure-execution-times-of-php-scripts) **AT4** – binary search. Possible scenario: locate the winning raffle ticket number in a digital competition.Using XML to load into an array* <http://blog.teamtreehouse.com/how-to-parse-xml-with-php5>
* <http://php.net/manual/en/simplexml.examples-basic.php>
 | Practice the use of binary search on arrays.Practice XML:* Write
* Load
* display
 |
| 4 | KK 5, 4, 14, 10, 11, 13 | InputTesting  |  **AT5** – xml load and display. May provide with xml file.HTML forms. Database + MySQL to connect, select data. Input data | Practice forms, selecting data from a DB, and displaying. Also inputting data to DB |
| 5 | \_\_\_\_\_\_\_\_\_AOS 2KK 17, 1 | \_\_\_\_\_\_\_\_\_\_\_Goals, objectives, data collection |  **AT6** – input data to mysql Database using a form.  – Test using suitable methods.* Provide students initially with layout diagrams, and goals (so they know what to test for)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Organisations’ goals, objectives. Instances in which data needs to be collected… techniques to collect data\* provide real world examples of solutns with presentation, logic, data layers\* | Decide on a real world need/opportunity that you wish to develop a solution for. |
| 6 | KK 2,3,4,5  | Constraint, scope, SRS | Constraint & scope of solutions. Analyse existing solutions.Discuss real-world needs or opportunities. Show sample SRS. Students to stickynote what they notice are key things in the document, or what would help them to build the solution if they received that SRS.Discuss student ideas **> Final idea due.** | Discuss your idea – ensure it meets the reqmts of the SAT.Begin writing SRS |
| 7 | KS 1, 2, 3KK 16 | Application architecture | Begin writing SRS within set time – scope, etc. SRS Parts 1, 2, 3Styles of modern application architecture  | writing SRS |
| 8 | KS 4 |  | Write the SRS Parts 4 to 7 | writing SRS |
| HOLS | 611 |  | Read: Techniques to generate design ideasFactors influencing the design of solutionsStart to flesh out ideas about how your solution will work – interface and database layers. We’ll learn more about how to clarify and present these in T2. |  |

**SAT Part 1**

* 9 weeks

**SAT Part 2**

* 8 weeks, plus
2 weeks holidays

Maybe adjust this so Part 2 has an extra week?

SAC 1

Six coding modules (ATs), demonstrating application of the skills & KK.

Will provide pseudocode, or IPOs or layout designs (AT6) with a mini scenario.

**SAT Part 1**

* SRS
* Alternate design ideas
* Detailed presentation of one design option
* Project management plan

**Software Dev – Term 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Key knwldg | Key skills | In class (teaching and learning tasks) | At home |
| TERM 21 | KK 12 |  | Interface and UX design, mockupsDiscuss design ideas with teacher, modify at home. | Modify design ideas. |
| 2 | KK 15 |  | Security influencing design – data protection, authentication…Use Case diagramsContext and data flow diagrams | Practice these design tools, apply to provided scenarios. |
| 3 | KK 8,9,10,13 |  | Data dictionary, pseudocode, object descriptionsNaming conventions (files, objects, internal documentation)Finish the SRS Parts 8 to 10 – Hand in. |  |
| 4 | KS 7 KK 1418 |  | Project Management – concepts and Gantt charts. How to use these during development also. Software: GanttProject, Word for WBS.Legal reqmts – ownership and privacy of data & info  | Practice making a Gantt chart for provided scenarioWork on project plan. |
| 5 |  |  | Project Plan due at start of weekSolution design - Propose and rough outline of 2-3 alternative designs. | Design ideas, final solution |
| 6 |  |  | Final solution Design – hand in at start of week. Teacher provides feedback/guidance if changes need to be made to the design prior to start of Development – this won’t impact marks for U3O1 but will set student up to not fail in the next part.  | Final solution design |
| 7 | U4 AOS 1 |  | Development |  |
| 8 |  |  | Development |  |
| 9 |  |  | Development |  |
| 10 |  |  | Development |  |
| 11 |  |  | *Buffer time(for sports days, excursions etc)* |  |
| Hols |  | Holidays | Development – require students to once a week submit/update teacher with screenshots and explanation of their work. Can be a video w voiceover, or images/posts to the class blog. |  |

**Software Dev – Term 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Week | Key knwldg | Key skills | In class (teaching and learning tasks) | At home |
| TERM 31 |  |  | Development |  |
| 2 |  |  | Development |  |
| 3 |  |  | Development |  |
| 4 |  |  | DevelopmentHand in | = 8 weeks + 2 weeks holidays |
| 5 | U4 AOS 2 |  | Begin U4 AOS 2… utilise flipped learning in the previous week to begin students on reading relevant sections of textbook. |  |
| 6 |  |  | “ |  |
| 7 |  |  | Practice SAC task, completed indiv, followed by class discussion and feedback. |  |
| 8 | U4 O2 |  | Final SAC – written report / response to questions / 1.5 hours estimate |  |
| 9 | Exam rev |  | Exam revision |  |
| 10 |  |  | *Buffer time(for sports days, excursions etc)* |  |