Advanced Internet Programming Assessment 2

Motivation

A professional developer does more than just write code. With enough persistence, *anyone* can use Google to copy-and-paste their way to an application that works. What sets a professional apart is a constant desire to learn, engage and improve at multiple levels:

- Creating high quality code
- Improving the processes of writing code
- Helping others to learn and improve
- Providing thoughtful insight and leadership
- Contributing to the profession

This assessment is motivated by learning objectives 1, 2, 3, 4 and 7 from the subject outline.

Assessment Item

For this assessment, you will prepare a portfolio demonstrating ongoing commitment to learning and professional development.

In Appendix A of this document there is a list of task choices. Over the course of the semester, you are required to complete **eight tasks**, comprising at least **one task from each category**.

As you complete the tasks, you will create a ‘portfolio’ of evidence. This is a document containing photos, code, URLs, text or other evidence that is required by the tasks.

The ‘Video presentation’ task **is mandatory and must be completed during or before Week 8** (ready for in-class discussion the following week).

You are expected to submit weekly from Week 2 onwards, but you can take up to three weeks absence or illness without the need for special consideration.

Submission

You will add to your portfolio each week using the UTS Online link. Any uploaded files should be in PDF, JPEG, PNG or Word format. Source code may be uploaded in a ZIP file or by linking to a public github/bitbucket repository.

For the purposes of this Assessment item, each week concludes at 11:59pm on Friday.

You are permitted to do multiple tasks (or all tasks!) in a single week if you wish to complete this assessment item early.

You are permitted to use comments to revise or add detail to previously submitted portfolio entries. The final state of your journal at 11:59pm Friday on November 2 will be used for marking.

Assessment Criteria

This assessment is worth 30% of your final grade and will be marked out of 30.

You will receive verbal feedback during the semester through in-class discussion, as well as detailed feedback with your final submission.

The final submission of your portfolio will be assessed according to the following criteria:

Demonstration of ongoing commitment to learning (6 marks)

0	You have not demonstrated an interest in ongoing learning or you have not completed the required number of tasks (eight tasks).
2	You have demonstrated an ability to improve your own learning as it relates to the learning objectives of the subject.
4	You have demonstrated a commitment to advancing your own mastery as required to maintain relevance in advanced internet programming.
6	You have demonstrated a 'relentless restlessness': a deep, independent, ongoing commitment to advancing your own mastery of the skills required to lead professional developers and your own mastery of the learning objectives of the subject.

Depth of understanding and insight into principles and technologies used in advanced internet programming (12 marks)

0	The evidence in your portfolio demonstrates mere familiarity with principles and technologies in advanced internet programming.

4	The evidence in your portfolio explains the principles and technologies used in advanced internet programming but only demonstrates superficial insights.
8	The evidence in your portfolio demonstrates insight. You have drawn novel connections between concepts and have identified compelling comparisons/contrasts in the principles and technologies.
12	The evidence in your portfolio demonstrates a deep understanding of the principles and technologies used in advanced internet programming. You have generated original insights into challenges for the field. You have expertly critiqued the application of the principles and technologies of advanced internet programming.

Ability to engage with and communicate professionally with technical peers (maximum 12 marks)

0	The evidence in your portfolio is confusing or uninformative.
4	The evidence in your portfolio is somewhat unclear or has minimal informative value. You have not demonstrated an ability to communicate at the minimum standards expected in professional technical contexts as it relates to internet programming.
8	The evidence in your portfolio is mostly clear and informative. You have demonstrated an ability to communicate at a professional level as it relates to internet programming.
12	The evidence in your portfolio is very clear, succinct and informative. You have demonstrated a mastery of professional technical communication as it relates to internet programming.

Assessment Penalties

It is an important requirement that you complete **at least one task from each category** in Appendix A. If you do not complete at least one task from every category, your submission is incomplete. In this case, the maximum grade for this assessment will be capped at 50% (i.e., 15 out of 30).

It is expected that you will complete items each week beginning Week 2. However, an allowance of three weeks for absence or illness has been included in this assessment.

If you submit less than eight tasks, you will be penalized by the Assessment Criteria for ‘Demonstration of ongoing commitment to learning’. In addition, please note that a submission which is missing many tasks is unlikely to contain sufficient evidence for you demonstrate the required skills and knowledge required by the other Assessment Criteria.

The table below indicates the last possible submission date for portfolio entries.

	Submission Due	**Last Possible Submission**
Week 1		
Week 2	First task	
Week 3	Second task	
Week 4	Third task	
Week 5	Fourth task	
Week 6	Fifth task	First task
Week 7	Sixth task	Second task
StuVac		
Week 8	**Video Presentation (Seventh task)**	**Video Presentation (Third task)**
Week 9	Eighth task	Fourth task
Week 10		Fifth task
Week 11		Sixth task
Week 12		Seventh task
StuVac		
Assessment Week 1		Eighth task

You are permitted to do several tasks in a week if you wish to complete this assessment early. Late tasks will be ignored. Completing extra tasks more than the minimum number required cannot make up for late submissions or an incomplete submission.

Absences and Special Consideration

You will **not** have an opportunity to resubmit work after the last possible submission date. However, there are 12 weeks and you will receive feedback throughout the period. You are permitted to miss three weeks due to illness, absence or any personal reasons without needing to provide evidence and still obtain full marks.

In the event of illness of one to three weeks, no special consideration is required (because three weeks of absences are already permitted).

For this assessment, special consideration will only be possible where there is evidence of significant illness (or similar circumstances) explaining a delay of more than three weeks.

Note that it is important to backup your work regularly. Special consideration will not be entertained for data loss.

Misconduct

These activities are to be completed by yourself during the semester. Claiming credit for something you have not done (i.e., plagiarism or fraud) or for work that you have completed prior to the start of semester or for another class (i.e., self-plagiarism) will be regarded as misconduct.

If you are in doubt, please ask your tutor. Please refer to the faculty's handbook for more information about Student Misconduct:
http://www.uts.edu.au/sites/default/files/FEIT%20Student%20Guide.pdf

Appendix A
Task Choices

You must choose tasks from the lists below. The tasks must be chosen so that they include at least one item from each of the five categories below. Other tasks are permitted but only with the prior written approval of your tutor or the subject coordinator (you can ask on UTS Online).

You cannot use the same task more than once.

Word counts are approximate.

All activities must be conducted during this semester. Claiming credit for something you have completed prior to the start of semester or for another class will be regarded as misconduct.

You can choose when to add the work to your portfolio. For example, you may complete multiple tasks in Week 1 but record it them in your portfolio in later weeks.

Hint: ‘Take an online course’ in Category 3 is a good choice to begin with if you have not used JavaScript before (i.e., watch a course on JavaScript or Node.js).

Reminder: ‘Video presentation’ in Category 1 must be completed in or before Week 8.

Category 1: Presentation and Communication

◿ **Video presentation**

By the end of Week 8, prepare a four minute video presentation on one of the emerging technologies in the list below. You should speak during the video and your face should be visible for at least part of the video. You do not need to edit or include visual effects - it is perfectly acceptable to just record yourself speaking in front of a computer screen.

Design your presentation so that it would help your classmates understand what the technology does and how it works. You should either include example code and/or a technical demo that you have prepared using the technology.

If you would like to research a topic not in the list below, you **must** obtain prior approval from your tutor.

[Portfolio: a link to your four minute video presentation on the technology (and upload the slides as a PDF/PPT if required)]

- *New features in ECMAScript 2017*
- *New features in ECMAScript 2018*
- *GraphQL (https://graphql.org/)*
- *AWS Lambda (https://aws.amazon.com/lambda/)*
- *Google Cloud Functions (https://cloud.google.com/functions/)*
- *Websockets (https://tools.ietf.org/html/rfc6455)*
- *Socket.io (https://socket.io/)*
- *Emscripten (http://kripken.github.io/emscripten-site/)*
- *asm.js (http://asmjs.org/spec/latest/)*
- *WebAssembly (https://webassembly.org/)*
- *Typescript (https://www.typescriptlang.org/)*
- *Flow (https://flow.org/en/)*
- *React Native (https://facebook.github.io/react-native/)*
- *Apache Cordova (https://cordova.apache.org/)*
- *Electron (https://electronjs.org/)*
- *Tensorflow.js (https://js.tensorflow.org/)*
- *Gulp (https://gulpjs.com/)*
- *Mocha (https://mochajs.org/)*
- *Chai (http://www.chaijs.com/)*
- *Nodemailer (https://nodemailer.com/)*
- *Custom Extensions for Visual Studio Code (https://code.visualstudio.com/docs/extensions/overview)*
- *Headless Chrome / Firefox Headless Mode*
- *Docker (https://www.docker.com/community-edition)*
- *Artillery.io (https://artillery.io/)*
- *RxJS (https://github.com/ReactiveX/RxJS)*
- FormatJS (https://formatjs.io/)

Category 2: Improving Quality and Productivity

◿ **Create automated tests**
Write meaningful unit tests for your Assessment 3. You might use a unit testing framework (such as Mocha or Chai) or you could use an functional test automation tool (such as Selenium).
[Portfolio: the source code (or a URL with the source code) of the relevant tests]

◿ **Conduct a user test**
Find a potential user for your Assessment 3. The user must not be a professional developer, must not be studying IT/computing and must not be a close friend. Have them use your application while you take notes. Do not speak while they are using the application. Just watch them and take notes, then ask your questions at the end. Ask where and why they struggled, understand how they would use your application in their life, learn as much as you can.
[Portfolio: a photo of your user using your application and 150 words summarizing key issues you learned about your Assessment 3 from the user test]

◿ **Reflect on and improve your focus**
Try the "pomodoro" method or another focus technique when writing code.
[Portfolio: 300 words explaining what happened and why the focus technique did or did not help you (it is fine if it didn't work for you—but analyze why)]

◿ **Reflect on your learning**
Consider some techniques that you can use to study more effectively in this subject or otherwise learn more about internet programming? Try some of the techniques over a few days and see if it makes a difference.
[Portfolio: 300 words describing what you tried and why it did or did not help you (it is fine if it didn't work for you—but analyze why)]

◿ **Reflect on your workflow**
What is one way that you can work more efficiently as an internet programmer? Try it. Consider doing something dramatic to see if you can make big improvements in your efficiency or productivity.
[Portfolio: 300 words discussing what you tried and what happened when you tried it]

◿ **Change your development tools**
Think about the different tools you are using as a developer (e.g., your operating system, your computer, your editor, your browser configuration). Change or customize those tools in order to improve your productivity or reduce errors.
[Portfolio: 300 words explaining what and why the changes improved (or did not improve) your productivity and what you would like to try next]

Category 3: Personal Development

- ◿ **Personal study**
  Read a classic book or research paper relevant to internet programming. Select a paper from one of the lists below.
  [Portfolio: 300 words describing what you have personally learned from the paper and how you might apply the knowledge in your Assessment 3 or elsewhere in your life]
  - ○ *https://en.wikipedia.org/wiki/List_of_important_publications_in_computer_science*
  - ○ *http://www.cs.cmu.edu/~aldrich/papers/classic/*
  - ○ *https://cstheory.stackexchange.com/questions/1168/what-papers-should-everyone-read*
  - ○ *http://blog.fogus.me/2011/09/08/10-technical-papers-every-programmer-should-read-at-least-twice/*
- ◿ **Solve the same problem twice**
  Try solving a problem using two or more different technologies or platforms not covered in class (e.g., deploy the same code to Amazon, Google Cloud and Azure and compare the differences in how you approach it) (e.g., try Angular 1 and Angular 2 on a small experiment and compare the two).
  [Portfolio: 300 words discussing what you tried and comparing/contrasting the technologies in relation to the specific problem you solved]
- ◿ **Take an online course**
  Complete at least 2 hours of an online course or training videos on a technology, skill or process relevant to this subject (e.g., look on Lynda.com, Udacity, Coursera or iTunesU).
  [Portfolio: Screenshot of the course progress and a 300 word summary of an new insight or connection that you gained as a result of studying the course]
- ◿ **Try a functional programming language**
  Create a small web application using one of the following functional programming languages: Haskell, Elm, Erlang, F#, Scala or Clojure.
  [Portfolio: your code and 300 words comparing/contrasting your experience doing web development in the functional programming language with your experience using JavaScript]

Category 4: Online Community

◿ **Answer a question on Stack Overflow**
Provide an *excellent* answer to a real question on Stack Overflow. Your answer doesn't need to be the accepted answer but it should be comprehensive and get some upvotes.
[Portfolio: the text of the answer and URL of the answered question on Stack Overflow and a screenshot from your logged in profile page showing that you answered the question]

◿ **Write a blog post**
Write a blog post on a topic relevant to internet programming. You can post to any public platform: if you don't have a personal homepage, consider a blogging platform such as medium.com.
[Portfolio: the URL linking to the blog as well as a copy of the text in the blog post]

◿ **Submit a commit, pull request or bug report to an open source project**
Contribute meaningfully to an open source project (i.e., a project started by somebody other than yourself). It doesn't need to be a groundbreaking contribution. It could be code, documentations or bug reports. It should be a change that improves the project in some small way. Your contribution should be clear, professional and well written.
[Portfolio: the URL of the contribution as well as the text or screenshot of the contribution (in case the contribution is deleted, resolved or ignored)]

◿ **Start an open source project**
Turn some code you have created into a reusable open-source library. It doesn't need to be sophisticated or fancy. It might be a project with just one file. However, it should be self-contained so that somebody else might want to use it in their own project. It should include a README or other documentation to help a new user. It should be a useful library or framework (not a complete application).
[Portfolio: the URL of the public repository]

Category 5: Face-to-Face Community

- ◿ **Attend a meetup or user group**
  Find and attend a meetup or user group relevant to internet programming. Hint: use meetup.com or search Google by combining a technology name with ‘user group Sydney’).
  [Portfolio: take a ‘Selfie’ at the event and write 200 words reflecting on something you learned during the event]
- ◿ **Attend a Hackathon**
  Participate in a Hackathon.
  [Portfolio: take a ‘Selfie’ at the event and write 200 words reflecting on something you learned during the experience]
- ◿ **Attend a conference or seminar**
  Attend a conference, seminar or workshop related to internet programming or software development.
  [Portfolio: take a ‘Selfie’ at the event and write 200 words reflecting on something you learned during the event]
- ◿ **Meet a professional developer**
  Meet a professional developer and ask them about their work or for their advice on your career, your resume, your development stack, your code or any other question relevant to your journey as a professional developer. The developer should not be a very close friend or somebody you have already had a similar conversation about before.
  [Portfolio: take a ‘Selfie’ with the developer and write 300 words describing who you met, what surprises you learned from talking with him/her and how their advice will change your own work or career]