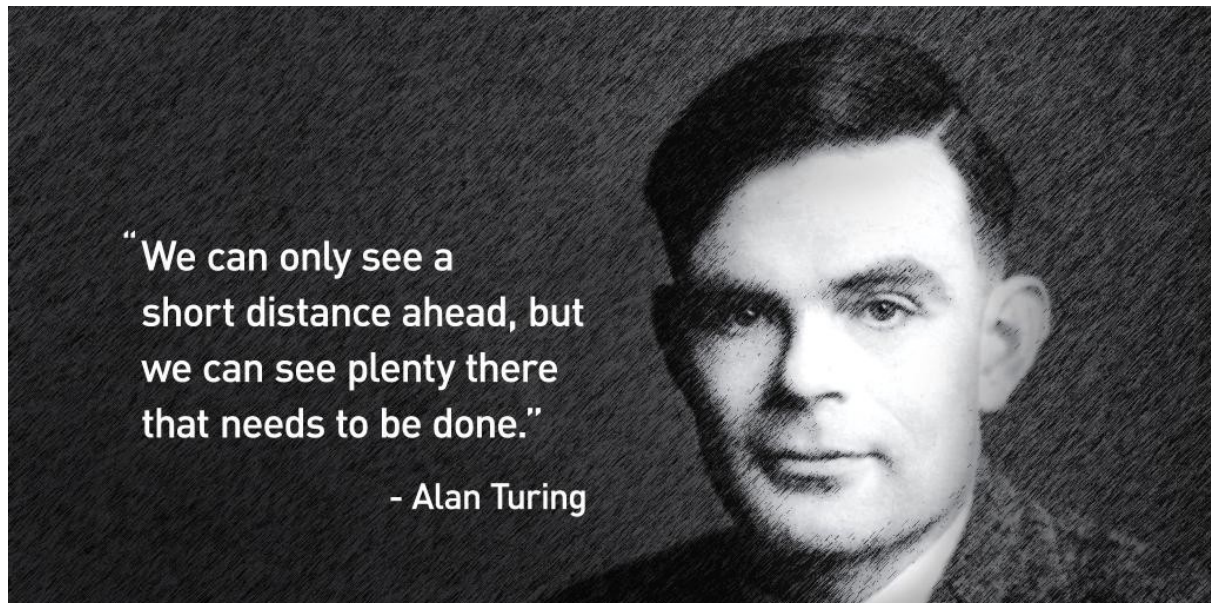


KIT101 Programming Fundamentals

Learning Reflection Report

Paul Watts (569987)



Self-assessment Details

The following checklists provide an overview of my self-assessment for this unit.

	PP	CR	DN	HD
Self-assessment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Self-assessment Statements

	Submitted/completed
Learning Reflection Report	<input checked="" type="checkbox"/>
Test 1 and Test 2 are Completed in MyLO	<input checked="" type="checkbox"/>
Programs that demonstrate coverage of core concepts	<input checked="" type="checkbox"/>

Minimum Pass Checklist

	Submitted/completed
All Credit Tasks are Completed on MyLO	<input checked="" type="checkbox"/>

Minimum Credit Checklist, in addition to Pass Checklist

	Submitted/completed
Distinction tasks (other than Custom Program) are Completed	<input checked="" type="checkbox"/>
Custom program meets Distinction criteria	<input checked="" type="checkbox"/>
Design report has structure chart and screenshots of program	<input checked="" type="checkbox"/>

Minimum Distinction Checklist, in addition to Credit Checklist

	Submitted/completed
Custom program meets HD requirements	<input checked="" type="checkbox"/>
Interview time booked	<input checked="" type="checkbox"/>
HD Project	<input checked="" type="checkbox"/>

Minimum High Distinction Checklist, in addition to Distinction Checklist

Portfolio Overview

This portfolio includes work that demonstrates that I have achieved all Unit Learning Outcomes for KIT101 Programming Fundamentals to a **High Distinction** level.

Through my Portfolio Tasks, my attendance at all Lectures and Tutorials, my participation in the Discussion Forum and my studies beyond what has been taught in this Unit, I believe I have demonstrated the focus, discipline, resilience and capabilities to achieve a High Distinction level.

Having started the Unit with zero knowledge of Java and almost zero knowledge of Object Oriented and Functional Programming my Portfolio tasks have demonstrated a continual improvement in my “big picture” understandings, my coding style and capabilities and the ability to continually learn from my mistakes.

I have done an extensive amount of wider reading and video tutorials stimulated by the material in this Unit, seeking out solutions and understandings to resolve the problems I encountered and to gain a better understanding of many of the concepts.

I devised a game plan at the start of the unit, supported by my software-based task management and knowledge management systems, to accomplish every Portfolio task available by the 16th of October and with the submission of this Learning Report within this deadline I will have achieved this goal.

Along the way I have built a vast repository of knowledge, not just with the Task Portfolio but also in additions to my own personal knowledge management system.

When I look back to the ubiquitous “Hello World” program in 1.2PP to completing all of the tasks today I can see a huge progression in my understanding of Java and the programming fundamental principles taught in this course.

The early tasks set the scene for ensuring proper programming conventions with a focus on good documentation and demonstrated good programming style.

This was continued throughout the course and I believe I not only understand the principles of how to do this but also the important reasons why we should do it.

A focus early on was how to read code and demonstrate this understanding by code tracing and this was further reenforced in the two tests during the semester, which I successfully passed.

From 4.3CR onwards we progressively encountered increasing complexity with the need to step back and think through a problem and solution and apply functional decomposition to break complex problems down into simple, elegant solutions with both a high degree of efficiency and code reuse.

I believe I demonstrated my understanding of this important concept in 4.3CR, 5.4DN, 6.2CR and especially in programming my own program design in 7.3CR, 7.4DN and 7.5HD.

When I look at any code now, functional decomposition comes to mind in terms of how it can be made simpler and easier to understand for humans (not just for computers).

While Object Oriented Principles started to be used in 2.1PP, 3.2PP, 3.3PP, 4.2PP and 5.2PP it wasn't really until 5.4DN and then especially from 6.1PP onwards that I really started to think in an OOP way. 7.1PP was especially challenging at the time but would not be now that I have had more

practice on implementing an OOP design (with my own program) and managing complexity in the later tasks.

I have not covered loops, data types, conditional statements, and learning the actual language syntax as believe these are foundational to any language and the concepts were well introduced in a progressive manner in almost lecture and task and I believe I have demonstrated, through all my Portfolio tasks a very good understanding of these learning outcomes.

What I have tried to focus on, through the lectures, course material, feedback from my Tutor, and my wider reading is the application of everything I have learned in a way that demonstrates best practice and an understanding of good programming style.

Reflection

The most important things I learnt

I could literally write pages under this point but will limit myself to just the top few:

1. I was surprised to learn, mainly from my wider reading and attending a number of Java related events, e.g. the launch of Java 15, that Java is still an extremely widely used language and continuing to be extensively improved. It is now 25 years old and used in billions of devices from Android phones to large scale systems running in huge corporations. This has made me excited to learn Java and continue learning it.
2. Working on my own program in 7.3CR, 7.4DN and 7.5HD and the feedback from my Tutor, really reenforced the importance of Functional Decomposition, and drove home OOP and Functional programming principles. I feel I am still very much the novice on OOP and Functional programming but very keen to continue my learning in these disciplines.
3. I learnt the benefits of a strongly typed language with the compiler able to resolve my intent at compile time, and not compile if there are errors, rather than experiencing the same problems at run time as with weakly typed languages.
4. Having not programmed for many years, other than doing various internet tutorials that didn't "stick", I was reminded that programming requires you to be extremely precise, logical and have the unflagging desire to solve puzzles. You also need to be constantly looking to do things in a better way, either more efficiently or in a simpler and more understandable (by humans) way.
5. Being a habitual procrastinator, I learnt the satisfaction and pleasure that comes from completing things on time, to a high standard and how to stop procrastinating.
6. Above all I learnt that personally I really need to code every day, preferably on projects that challenge me to the next small level of knowledge, otherwise the knowledge is not retained. It's definitely not like riding a bike. The lectures, portfolio tasks and tutorial sessions have been excellent in this regard as they have bought me along gradually to those next levels of understanding while maintaining my strong interest in doing so. I will continue this example with increasingly more challenging projects after this Unit ends.
7. Having run my little Java self- learning programs on my Windows PC, My MacBook Pro and my Raspberry Pi 4 , I've learnt the amazing benefit of the JVM and the "write once, run anywhere" ethos of Java. Very excited to learn about executable .jar files and have them working as part of this experimentation. Also learnt about packages and modules.
8. Being my 1st time at University and KIT101 my first University subject I wasn't sure what to expect or what I would be learning. I've been very pleasantly surprised by this Unit and feel it has been a great start to my University education.

The things that helped me most were

1. Interesting and entertaining lectures that continued to build on previous knowledge learnt. James enthusiasm, humour and practical code demonstrations turned on many light bulbs.
2. Quality reading materials, both unit focused and links to wider readings. This led me in many directions on my wider study that I will continue to pursue.
3. Having good code examples that I could download and play with and learn from. Also having the starter code to a number of the portfolio tasks no doubt saved a lot of frustration.
4. Found the assistance from Amanda invaluable and very encouraging. Definitely going to miss having someone review my work and point me in the right direction on how to improve my code.

I found the following topics particularly challenging

1. Literally everything was challenging at Day 1 and probably for a few weeks afterwards.
2. At first I found the Java Class/Method/Variable Modifiers challenging as to when and why they are used. I put together a "Cheat Sheet" that made this more understandable.
3. The whole concept of OOP was very challenging and I ended up doing a 3rd party video course plus reading a large number of articles to come to grips with it mid-way through the Semester. The Refactoring in 8.1DN was a big help in putting OOP and the "Driver" "Organiser" model into practice and I then refactored my own program in 7.4DN (which I had basically completed) into this model.
4. Recursion was definitely very challenging and when I first stuck a method with the same name as well as recursion in an earlier portfolio task it totally threw me. The final lecture on Recursion was very helpful in understanding the concept.

I found the following topics particularly interesting

1. Although very challenging at first the whole concept of OOP, and it's four basic principles of Encapsulation, Inheritance, Abstraction and Polymorphism very interesting and something I wish to master over time.
2. Although not specifically taught in this Unit, it did stimulate me to start reading about data structures and algorithms which I'm definitely interested in exploring further and realise a good knowledge of all the basic structures and patterns will be a great help to be a good programmer in any language.
3. I found the concepts and application of Functional Decomposition very interesting and something I wish to continue to practice in my programming.

I believe I learned these topics, concepts, or tools really well

1. I learnt the IntelliJ IDEA IDE very well and had a lot of practice debugging my code in it.
2. I learnt the concepts that a class is a blueprint to instantiating an object well. This was a very foreign concept to me at first.
3. Learnt to read Java 8 code well and can follow the execution flow in my mind as well as using Code Tracing.

I still need to work on the following areas

1. I have a novice understanding of OOP after this course but really need to do a lot more Projects to master this. Have this planned after the Semester ends.
2. Don't have a good grasp on when to use Functional Programming and when to use OOP.

My progress in this unit was ...

1. From 2.1PP to 3.3PP it was very much a dive into the deep end and the first time encountering the Turtle was a bit of "what is going on here".
2. 3.4PP to 5.3PP were fairly straight forward for me but 5.4DN was a challenge due to a number of new concepts, e.g. methods with the same name and recursion in one of them. This threw me for a while, but I got a lot of satisfaction from completing the program successfully.
3. The first forays into OOP in 6.1PP and 6.2CR was both interesting and challenging and led me to doing additional Internet tutorials to understand the OOP principals better.

4. 7.1PP was definitely a slow time as I encountered a lot of new challenges and concepts to grasp.
5. The move into working on my own program, begun in 7.3CR and culminating in 7.5HD was definitely the most challenging but most rewarding of the whole course. An inordinate amount of time was spent on understanding how things worked and encountering problem after problem but also enormous satisfaction in figuring it out and getting it working in the end. Applying the learning outcomes from the course to my own program provided the most accelerated learning in the Semester.

This unit will help me in the future

1. It is already helping me as I can now look at code in other languages and have some idea of what it is doing. I also have the desire to learn more languages, e.g. Dart (and Flutter), Rust, Python, JavaScript (TypeScript) and Swift.
2. Also, very keen to continue my studies in Java as from my wider reading understand we have really just scratched the surface in this course.
3. It has reignited my passion for Software Development and crystallised my decision to take this as my major.

If I did this unit again I would do the following things differently

1. I would try and get to grips much earlier than I did with what classes, methods, objects, class libraries, method signatures, and access and non-access modifiers actually are and what they mean and not just how they are declared.
2. I would learn debugging techniques in the editor much earlier to assist me in understanding the code and resolving errors. It's almost magical that you can jump (follow your code execution) into the Java libraries with the debugger to see exactly what is happening (although it will be a long while before I fully understand it).
3. Other than these two things I worked to a game plan for the Course that I believe I executed well.

Other comments

1. I believe the successful completion of every Portfolio task available (with the HD tasks and this Learning Report to be reviewed and marked completed) is an adequate demonstration of my understanding and putting into practice the Learning Outcomes for this Unit.
2. I believe in terms of Java and OOP I now know what I don't know and have plans to continue to peel away the onion skin of knowledge, layer by layer, to progressively fill in these knowledge gaps.
3. I've learnt that the only way to really learn Programming and retain the knowledge is by doing rather than by reading or watching lectures and tutorials. The Portfolio tasks were a great example of this and I have decided that I will continue to code every day so the skills I have learned will not be wasted.