PDA: Software Development
Level 8 - Student Evidence Checklist

Full name	Ian Henderson	**Key:**	A & D - Analysis and Design Unit I & T - Implementation and Testing Unit P - Project Unit
Cohort	E17		

The evidence required can be taken from your assignments, homework that you have completed on your own or by creating a specific example for the PDA.

Week 2	Unit	Ref.	Evidence	Done
	I & T	I.T 5	Demonstrate the use of an array in a program. Take screenshots of: ● An array in a program ● A function that uses the array ● The result of the function running	X
	I & T	I.T 6	Demonstrate the use of a hash in a program. Take screenshots of: ● A hash in a program ● A function that uses the hash ● The result of the function running	X

Week 3	Unit	Ref.	Evidence	Done
	I & T	I.T 3	Demonstrate searching data in a program. Take screenshots of: ● Function that searches data ● The result of the function running	X
	I & T	I.T 4	Demonstrate sorting data in a program. Take screenshots of: ● Function that sorts data ● The result of the function running	X

Week 5	Unit	Ref.	Evidence	Done
	A & D	A.D 1	A Use Case Diagram	X
	A & D	A.D 2	A Class diagram.	X
	A & D	A.D 3	An Object diagram.	X
	A & D	A.D 4	An Activity Diagram	

	A & D	A.D 6	Produce an Implementations Constraints plan detailing the following factors: ● Hardware and software platforms ● Performance requirements ● Persistent storage and transactions ● Usability ● Budgets ● Time	
	P	P 5	Create a user sitemap.	X
	P	P 6	Produce two wireframe designs.	X
	P	P 10	Take a screenshot of an example of pseudocode for a function.	X
	P	P 13	Show user input being processed according to design requirements. Take a screenshot of: ● The user inputting something into your program ● The user input being saved or used in some way	
	P	P 14	Show an interaction with data persistence. Take a screenshot of: ● Data being inputted into your program ● Confirmation of the data being saved	
	P	P 15	Show the correct output of results and feedback to user. Take a screenshot of: ● The user requesting information or an action to be performed ● The user request being processed correctly and demonstrated in the program	
	I & T		Coding exercise 1: Static and Dynamic testing task A	X

Week 6	**Unit**	**Ref.**	**Evidence**	**Done**
	I & T	I.T 7	Demonstrate the use of Polymorphism in a program.	X

Week 9	**Unit**	**Ref.**	**Evidence**	**Done**
	A & D	A.D 5	An Inheritance Diagram	
	I & T	I.T 1	Take a screenshot of an example of encapsulation in a program.	X
	I & T	I.T 2	Take a screenshot of the use of Inheritance in a program. Take screenshots of: ● A Class ● A Class that inherits from the previous class ● An Object in the inherited class ● A Method that uses the information inherited from another class.	X

	P	P 11	Take a screenshot of one of your projects where you have worked alone and attach the Github link.	
	P	P 12	Take screenshots or photos of your planning and the different stages of development to show changes.	

Week 11	Unit	Ref.	Evidence	Done
	P	P 18	Demonstrate testing in your program. Take screenshots of: ● Example of test code ● The test code failing to pass ● Example of the test code once errors have been corrected ● The test code passing	

Week 12	Unit	Ref.	Evidence	Done
	P	P 16	Show an API being used within your program. Take a screenshot of: ● The code that uses or implements the API ● The API being used by the program whilst running	

Week 14	Unit	Ref.	Evidence	Done
	P	P 1	Take a screenshot of the contributor's page on Github from your group project to show the team you worked with.	
	P	P 2	Take a screenshot of the project brief from your group project.	
	P	P 3	Provide a screenshot of the planning you completed during your group project, e.g. Trello MOSCOW board.	
	P	P 4	Write an acceptance criteria and test plan.	
	P	P 7	Produce two system interaction diagrams (sequence and/or collaboration diagrams).	
	P	P 8	Produce two object diagrams.	
	P	P 9	Select two algorithms you have written (NOT the group project). Take a screenshot of each and write a short statement on why you have chosen to use those algorithms.	
	P	P 17	Produce a bug tracking report	
	I & T		Coding Exercise 2: Unit and Integration testing task B	