**7 PDA: Software Development**

**Level 8 - Student Evidence Checklist**

|  |  |  |
| --- | --- | --- |
| **Full name** | **Peter McCready** | **Key:** A & D - Analysis and Design Unit  I & T - Implementation and Testing Unit  P - Project Unit |
| **Cohort** | **G3** |

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   As part of Test Driven Development (TDD) I used an array within the assert\_equal function as the expected result of running the test\_array\_of\_capitals method.     * A function that uses the array   The array\_of\_capitals function used the capitals array, used in conjunction with country hashes in the TDD above. The empty capitals array is populated with the capitals of each country when the function is run.     * The result of the function running   When the function is run the value of ‘actual’ is set to the value held against the captial key in the country hash, for each country. |  |
| I & T | I.T 6 | Demonstrate the use of a hash in a program. Take screenshots of:   * A hash in a program   The @customers array holds each customer as a hash containing name, pets and cash keys with their associated values.     * A function that uses the hash   The function below sells a pet to the first customer (Craig) held in the customers array above. A find method is used to locate a specific pet and the sell\_pet\_to\_customer function is used to move that pet into the customer’s pets array which is held in the customer’s hash. The test below then checks to ensure the length of the pets array in the customer’s hash has increased to 1, checks that the pets\_sold by the pet\_shop is increased to 1 and checks the total cash of the pet\_shop has increased by the value of the sold pet (from 1000 to 1900, an increase of 900, the price of Arthur the Husky).     * The result of the function running   When the function is run the pet object is added to the customer hash’s pets array. |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week 3** | **Unit** | **Ref.** | **Evidence** | **Done** |
| I & T | I.T 3 | Demonstrate searching data in a program. Take screenshots of:   * Function that searches data   Combining the SqlRunner and self.find() method of the Artist class it was possible to find a specific instance of the Artist class. This was done by calling the method against the class in Pry-Byebug.       * The result of the function running   When the function was run, passing in the id of the Artist I was trying to find (27) the artist with that id is returned.    This can be cross-referenced with the artists table of the music database to confirm the correct data is returned: |  |
| I & T | I.T 4 | Demonstrate sorting data in a program. Take screenshots of:   * Function that sorts data   Combining the SqlRunner and self.all() method of the Album class it was possible to find all instances of that class. This was done by calling the method against the class in Pry-Byebug. ORDER BY was used within the SQL to sort the returned dataset.       * The result of the function running     To confirm that sorting has taken place this can be confirmed with checking against the entries held in the album table of the music database: |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week 5** | **Unit** | **Ref.** | **Evidence** | **Done** |
| A & D | A.D 1 | A Use Case Diagram |  |
| A & D | A.D 2 | A Class diagram. |  |
| A & D | A.D 3 | An Object diagram. |  |
| 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. |  |
| P | P 6 | Produce two wireframe designs. |  |
| P | P 10 | Take a screenshot of an example of pseudocode for a function. |  |
| 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   Changing the pricing details for Alter Bridge The Last Hero album:  Original values on edit album:    Changed ‘Buy Price’, ‘Discount’ and ‘Initial Sell Price’ before hitting ‘Amend Details’:    Note: ‘Final Sell Price’ and ‘Profit Margin’ are calculated after the ‘Amend Details’ button is clicked so is changed on the next page, see next screenshot below.   * Confirmation of the data being saved   Changed data saved and the ‘Final Sell Price’ and ‘Profit Margin’ calculated and displayed by the system: |  |
| 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   Album shown in list of Artists/Albums (All Stock) and request to ‘Delete Album’ (Dead Kennedys, Give me death):       * The user request being processed correctly and demonstrated in the program   Deleted Album removed from list of Artists/Albums (All Stock): |  |
| I & T |  | Coding exercise 1: Static and Dynamic testing task A  Uploaded to GitHub in Static and Dynamic Testing Folder. |  |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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. |  |
| 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. |  |
| 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 |  |