Fill the following table. For test input, expected output and output obtained, add as many cases as you have tested.

*See an example of a completed form in Blackboard.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Self-assessment  (select one) | Test input | Expected output | Output obtained | Comments |
| 1 | Fully implemented and working  Partially implemented  Not attempted | 1: | 1: Welcome to the New Theatre | 1: Welcome to the New Theatre | Correct |
| 2 | Fully implemented and working  Partially implemented  Not attempted | 1:  2: Option 0 selected | 1: Menu with 9 options  2: Program ends | 1: Menu with 9 options  2: Program ends | Menu working correctly |
| Insert screenshot of your menu here including the welcome message: | | | | | | | |
| 3 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 1 + row = 4, seat =10  2: Option 1 + row = a  3: Option 1 + row 1, seat 21  4: Option 1 + row 1, seat ABC | 1: An error message output  2: An invalid message output  3: An invalid message output  4: An invalid message output | 1: Shows ‘Error: This row number does not exist. Please select 1-3.’  2: Displays Invalid row number, please try again.  3: Displays an invalid message  4: Displays an invalid message | Working correctly |
| 4 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 1 + row = 3, seat = 10 + option 2  2: Option = a  3: Option = 10  4: Option = 2 | 1: Shows seating area with correct seat taken  2: Invalid message  3: Invalid message  4: Prints the seats accordingly | 1: Shows seating area with correct seat taken  2: Displays “Invalid option, please try again”  3: Displays “Invalid selection. Please choose a valid option from the menu above.”  4: Displays the seating area | Working correctly |
| Insert screenshot of your output for task 4 here after buying a ticket for row 1, seat 1 and row 3, seat 20 : | | | | | | | |
| 5 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 3 selected. User enters a row and seat that is not booked  2: Option 3 selected. User enters a row and seat that is booked | 1: Shows that the seat is not booked  2: The seat is cancelled | 1: Displays “This seat is available, do you want to book it now?”  2: Displays “The ticket has been successfully removed from the booking system.” | Working correctly |
| 6 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 4 selected. | 1: Displays the available seats correctly | 1: Displays the available seats correctly | Working correctly |
| 7 | Fully implemented and working  Partially implemented  Not attempted | 1: Before Option 5 I tried Option 6.  2. Option 5 is selected | 1: File not found error  2: Creates a file and get stored | 1: Displays “File not found”  2: Displays “File saved successfully” | Working correctly |
| 8 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 6 is selected | 1: If file not found it creates a file and loads into the file correctly | 1: Displays the data in 1s and 0s | Working correctly |
| 9 | Fully implemented and working  Partially implemented  Not attempted |  | A person instance created with the constructor is called | Person created with the given details | Working correctly |
| 10 | Fully implemented and working  Partially implemented  Not attempted |  | A ticket instance created with the constructor is called. | Ticket created with the given details | Working correctly |
| 11 | Fully implemented and working  Partially implemented  Not attempted |  | Method prints the ticket information when called | Method prints the details accordingly | Working correctly |
| 12 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 1 selected and then asks for user info.  2: Ticket prices validated so only 10, 20 and 30 can be entered  3: Ticket price = a or ticket price = 2.5  4: Email is validated, accepts only if the email contains @ and .  abc@acb.com  5: Email = abc  Email = abc@def  Email = abc.  6: Program asks for the inputs repeatedly until the correct one is entered  7: Option 3 | 1: Asks for name, surname, and email  2: Works fine it asks for the row number next  3: Displays an error message  4: Works fine asks for ticket price next  5: Displays and error message  6: Works correctly  7: Removes ticket from array list and works correctly with validations | 1: Asks for name, surname, and email  2: Works fine it asks for the row number next  3: Displays an error message  4: Works fine asks for ticket price next  5: Displays and error message  6: Works correctly  7: Removes ticket from array list and works correctly with validations | Working correctly |
| 13 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 7 is selected | 1: Prints ticket information and total price | 1: Prints ticket information and total price correctly | Working correctly |
| 14 | Fully implemented and working  Partially implemented  Not attempted | 1: Option 8 is selected | 1: Prints ticket information in the order of ticket price | 1: Prints ticket information in the order of ticket price correctly | Working correctly |
| 15 | Explain which testing strategy did you take (e.g., how you tested that the output is correct, different inputs, different values, wrong values, etc.)  I tried all the possible combinations of errors that could be in each and every line to check the error messages and when entered the correct one checking the expected output | | | | | |
| 16 | Did you include comments in your code? Is your code idented? Did you use your own functions? Are your variable names informative?  I have included comments in most areas that I think are important and when someone else is reading my code for better understanding.  My code is indented.  I have used functions that I’ve defined.  The variable names are relevant to the data that is stored. | | | | | |

Are there any parts of the coursework which you would like to get feedback?

|  |
| --- |
| I would like to get feedback on the efficiency of my code. |

**DEMO: You will have to demonstrate your understanding of your code during a tutorial (week 10 or 11). Remember to reference any websites, or technologies that you used in this coursework. Tasks 9-15 will not be marked if you do not attend the demo.**