Contact.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Type | Comments |
| Create a contact object and print the details. Data:  First Name: Kelvin Last Name: Chan Email: ycc1g11@soton.ac.uk | Correct result printed on screen | As expected | Normal | Using toString() method |
| Create a contact object with some null values and try to print the details | Correct details printed on screen. An empty string will be printed for those attributes which are null during construction | As expected | Error |  |

AddressBook.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Type | Comments |
| Create two contacts and add to a new AddressBook object | Successful message printed on screen | As expected | Normal |  |
| Add an existing contact to the AddressBook object | Failure message printed on screen | As expected | Error |  |
| Remove an existing contact from the AddressBook object by Full name | Successful message printed on screen | As expected | Normal |  |
| Remove an existing contact from the AddressBook object by Email | Successful message printed on screen | As expected | Normal |  |
| Remove an not existing contact from the AddressBook object by Full name or Email | Failure message printed on screen | As expected | Error |  |
| Search contacts by first name and last name | Returning a list of contacts with the required first name or last name | As expected | Normal |  |
| Search contacts by first name and last name which does not exist | Returning a empty list | As expected | Error |  |

Obstacle.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Type | Comments |
| Create a obstacle with proper inputs:  Name: “Boeing 777”  Height: 25  Then print its details by using the getters. | Printing the values given in the initialization | As expected | Normal |  |
| Create a obstacle with improper inputs:  Name: null  Height : 25  Then print its details by using the getters | Printing the height given in the initialization.  The name will be assigned as “Temp Obstacle” | As expected | Error |  |
| Create a obstacle with improper inputs:  Name: “Boeing 777”  Height : -5000  Then print its details by using the getters | Printing the name given in the initialization.  The height will be assigned to the default value 25 | As expected | Error |  |
| Create a obstacle with improper inputs:  Name: null  Height : -5000  Then print its details by using the getters | The name will be assigned as “Temp Obstacle”  The height will be assigned to the default value 25 | As expected | Error |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Type | Comments |
| Use setters to update the name, height, width and length with proper values and print them out:  Name: “Boeing 777”  Height : 30  Width : 400  Length : 1000 | Printing the values given to the setters | As expected | Normal |  |
| Use setters to update the name, height, width and length with improper values and print them out:  Name: null  Height : -500  Width : -1000  Length : -99999 | All of them will not be changed due to null or negative input. Original values printed on screen | As expected | Error |  |

PhysicalRunway.java

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Type | Comments |
| Add two Runway which defined properly to a PhysicalRunway class during initialization. Use getter to get them back and print their details | Correct details printed on screen | As expected | Normal |  |
| Assign null values as the Runway parameters of a PhysicalRunway constructor. Use getter to get them back and print their details | Default values of a Runway object with a name “L” or “R”, 3900 for TORA, TODA, ASDA, 3500 for LDA and 0 for displaced threshold will be assigned to the PhysicalRunway object if the parameter is null. | As expected | Error |  |
| Get the calculation through toCalculation method without placing an obstacle | Default values of the runway will be returned as there is no obstacle on the runway | As expected | Normal |  |
| Get the calculation through toCalculation method after placing an obstacle | New parameters and the full calculations will be printed on screen | As expected | Normal |  |
| Get the calculation through toCalculation method after placing an obstacle but the distance between the obstacle and center line is longer than 150 meters | Default values of the runway will be returned as it’s safe to use the whole runway | As expected | Normal |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input | Expected Output | Actual Output | Type | Comments |
| Get the calculation through toCalculation method removing an obstacle from the runway | Default values of the parameters will be printed on screen | As expected | Normal |  |
| Modify the values to make the calculations for the new parameters are negative:  Obstacle height : 100m  Runway original  TORA, ASDA, TODA : 3900  LDA : 3500  Distance away from threshold : 10m  Distance away from center line :  30m | Zero will be assigned as the new TORA ASDA, TODA and LDA if the new calculation results are negative. | As expected | Normal |  |
| Execute the setters for ID, Runway Strip width, RESA, stopway, blast allowance and angle of slope with null or negative values | Null or negative inputs will be ignored. Value unchanged | As expected | Error |  |
| Place an obstacle, for the landing over the obstacle and taking off towards the obstacle calculation, the multiplication of the angle of slope and height of obstacle is less than the default RESA. Print the calculations. | The multiplication result within the calculation will be replaced by the RESA value | As expected | Normal |  |