28/11/16

Testing Plan

Subject Records Application

KAMALJEET KARWAL (000845146)

# Unit Tests

## Address Class

* Testing Address class’s public methods.
* Test file name: AddressTest.java

### Variables used:

* Address default\_Address
* Address m\_address
* Address cloneAddress

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature**  **to be tested** | Set oftesting values | **Expected**  **outcome** | **Obtained?**  **(Y/N)** |
| Default constructor | default\_address  = new Address(); | default\_address =  [1,  Unkown St,  Unknown,  State,  0000] | Yes |
| Constructor with street  number  Street name,Suburb,  state and  Post code as parameters | mAddress = new Address (  "204","Unley Road","Unley","SA","5061"  ) | mAddress = [  204, Unley Road, Unley,  SA  ,5061] | Yes |
| Cloning Constructor | Address cloneMAddress = new Address(mAddress); | cloneAddess =[  204, Unley Road, Unley,  SA  ,5061] | Yes |
| Setters | defaultAddress.setStreetNo("U4/104")  defaultAddress.setStreetName  ("Norseman Ave")  defaultAddress.setSuburb("Hillcrest")  defaultAddress.setState("SA")  defaultAddress.setPostCode("5086") | defaultAddress[  U4/104,  Norseman Ave,  Hillcrest,  SA  5086] | Yes |
| Post code input  validity | defaultAddress.setPostCode("WDKI")  defaultAddress.setPostCode("10395") | Error Message  Sets post code = 0000  Error Message  Sets post code = 0000 | Yes  Yes |
| Getters | defaultAddress.getStreeNo()  defaultAddress.getStreeName()  defaultAddress.getSuburb()  defaultAddress.getState()  defaultAddress.getPostCode() | U4/104,  Norseman Ave,  Hillcrest,  SA  5086 | Yes |
| Static method  isPostCodeValid() | Address.isValidPostCode("5iuut") | False | Yes |
| Testing toString() | defaultAddress.toString() | U4/104,  Norseman Ave,  Hillcrest, SA,5086 | Yes |

## Lecturer Class

* Tests all the methods in the Lecturer class.
* Test file name: LecturerTest.java
* Id of Lecturer always start from zero and incremented every time a new Lecturer is created.
* Expected outcome is tested using toString Method

### Variables:

Lecturer lect\_a

Lecturer lect\_b

Lecturer lect\_c

Lecturer lect\_cloneA

address\_tafe = ("137","Days Road","Regency Park","SA","5010")

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature**  **to be tested** | Set oftesting values | **Expected**  **outcome** | **Obtained?**  **(Y/N)** |
| Default constructor | Lecturer lect\_a = new Lecturer() | ID:0  Jon Snow  Ph: (08)80008000  Address: 1, Unkown St, Unknown, State,0000 | Yes |
| Constructor with  First name,  Lastname  Phone number  And Address variable | lect\_b = new Lecturer  ("John","Appleseed",  "4056049564",address\_tafe) | lect\_b = [  ID:1  John Appleseed  Ph: 4056049564  Address: 137, Days Road, Regency Park,  SA,5010] | Yes |
| Constructor with  First name,  Lastname  Phone number  Street No, Street Name,  Suburb,State and  Post code as paramters | lect\_c = new Lecturer(  "Tom",  "Contoso",  "0880800808",  "1","Days Road"  ,"Regency Park",  "SA",  "5010"); | lect\_c = [  ID:3  Jon Snow  Ph: (08)80008000  Address: 1, Unkown St, Unknown, State,0000 | Yes |
| Cloning Constructor | lect\_cloneA =  new Lecturer(lect\_a); | lect\_cloneA =[  ID:3  Jon Snow  Ph: (08)80008000  Address: 1, Unkown St, Unknown, State,0000 | Yes |
| Setters | lect\_a.setFirstName("Bill");  lect\_a.setLastName("Gates");  lect\_a.setAddress(address\_tafe);  lect\_a.setPhone("060588900"); | lect\_a = [  ID:0  First Name = Bill  Last Name = Gates  Ph: 060588900  Address: 137, Days Road, Regency Park, SA,  5010 ] | Yes |
| Getters | lect\_a.getId()  lect\_a.getFirstName()  lect\_a.getLastName()  lect\_a.getFullName()  lect\_a.getPhone()  lect\_a.getAddress()  lect\_a.toString() | 0  Bill  Gates  Bill Gates  060588900  137, Days Road, Regency Park, SA,5010  ID:0  Bill Gates  Ph: 060588900  Address: 137, Days Road, Regency Park, SA,5010 | Yes  Yes  Yes  Yes  Yes  Yes  Yes |
| Static method  getCount | Lecturer.getCount() | 4 | Yes |

## Time Class

* Testing all methods in Time Class
* Test file name: TimeTest.java

### Variables Used:

* Time t1
* Time t2
* Time t3
* Time t4
* Time t5

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature**  **to be tested** | Set oftesting values | **Expected**  **outcome** | **Obtained?**  **(Y/N)** |
| Default constructor | t1 = new Time()( | Sets Time= 12:00 AM | Yes |
| Constructor with illegal  hour | t2 = new Time(24,45) | Sets hour = 0  Sets min = 24  Prints error message  Sets time = 12:45 | Yes |
| Constructor with  Illegal mins | t3 = new Time(23,61) | Sets hour = 23  Set min = 00  Prints error message  Sets time = 11:00PM | Yes |
| Testing valid time | t4=new Time (0,30) | Sets Time = 12:30AM | Yes |
| Testing negative values  Of hours and min | t5=new Time(-12,-30) | Sets Time = 12:30PM | Yes |
| setTime(hours,min)  Time setter using any  Values of hours and min  are hours and mins  from mid night (0:00). | t5.setTime(0,61)  t1.setTime(47,39) | Sets time = 01:01AM  Sets Time = 11:39PM | Yes  Yes |
| setHour(hour)  setMin(min) | t2.setHour(23)  t2.setMin(59) | Sets hour = 23  Sets Min = 59 | Yes  Yes |
| Testing Getters | t2.getHour()  t2.getMin()  t2.toString()  t2.toStringHrs() | 23  59  11:59 PM  23:59 Hrs | Yes  Yes  Yes  Yes |

## Subject Class

* Testing all the methods in Subject Class
* Test file name = SubjectTest.java
* Testing setters using toString() method.

### Variables used:

**\*Test Address**

address\_tafe = ("137","Days Road", "Regency Park","SA","5010")

**\*Test Lecturers**

Lecturer garyM = new Lecturer("Gary","Lion","080808080",address\_tafe);

Lecturer msJ = new Lecturer("Ms","Java","99808342", address\_tafe);

**\*Test Start Time**

Time oneThirty = new Time(13,30);

**\*Test Duration**

Time threeHours = new Time(3,0);

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature**  **to be tested** | Set oftesting values | **Expected**  **outcome** | **Obtained?**  **(Y/N)** |
| Default constructor | e126 = new Subject() | Sets e126 = [  CRN: 18000  Name: n/a  Lecturer: Jon Snow  Room: n/a  Start Time: 12:00 AM  Duration: 00:00 Hrs  ] | Yes |
| Constructor with Subject  Name, Lecturer, Address,  Room, start Time,  duration) | d111 = new Subject("Programing Solutions",  msJ,  "G29",  oneThirty,  threeHours); | Sets d111 = [  CRN: 18001  Name: Programing Solutions  Lecturer: Ms Java  Room: G29  Start Time: 01:30 PM  Duration: 03:00 Hrs  ] | Yes |
| Testing getFinishTime() | d111.getFinishTime() | 4:30 PM | Yes |
| Testing Setters | e126.setName  ("Engineering mathematics")  e126.setLecturer(garyM)  e126.setRoom(“G29)  e126.setStartTime  (new Time(9,30))  e126.setDuration(new Time(3,0)); | E126 = [  CRN: 18000  Name: Engineering mathematics  Lecturer: Gary Mate  Room: G29  Start Time: 09:30 AM  Duration: 03:00 Hr  ] | Yes |
| Testing Getters | Subject.getCrnCount()  e126.getCrn()  e126.getName()  e126.getLecturer()  e126.getRoom()  e126.getStartTime()  e126.getDuration() | 18002  18000  Engineering mathematics  ID:0  Gary Mate  Ph: 080808080  Address: 137, Days Road, Regency Park, SA,  5010  G29  9:30 AM  3:00 Hrs | Yes  Yes  Yes  Yes  Yes  Yes  Yes |

# SubjectApp Final Tests

File name SubjectApp.java

The application is required to do the following:

* Add a new Subject to the list
* Modify the Subject details of any Subject based on the CRN number
* Display all the details of all the subject in the list
* Display list of Lecturers and their phone numbers
* Display list of all the Subjects and their CRNs

### Variables used:

### String test\_phone = "(08)80800800"

* Address test\_address = new Address("137", "Days Road", "Regency Park", "SA", "5010")
* Lecturer test\_lecturer = new Lecturer("Joe", "Test", test\_phone, test\_address)
* Lecturer billG = new Lecturer("Bill", "Gates", test\_phone, test\_address)
* Lecturer steveJ = new Lecturer("Steve", "Jobs", test\_phone, test\_address)

When the app starts, it displays a main menu and ask for user input.

Testing the validity of the input below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature**  **to be tested** | Set oftesting values | **Expected**  **outcome** | **Obtained?**  **(Y/N)** |
| **Main menu input tests**  Displays a following menu for the user and asks for input.  \*\*\*\*\*\*\*\*\*\*\*\*\*\* MAIN MENU \*\*\*\*\*\*\*\*\*\*\*\*\*\*  1. Add a subject to the list.  2. Modify Subject details.  3. Display all Subjects with all details.  4. Display Subjects with CRNs.  5. Display Lecturers and their phone numbers.  6. Exit | | | |
| Character input (invalid) | a | Error Message  Re-ask for input | Yes  Yes |
| Negative number input | -1 | Error Message  Re-ask for input | Yes  Yes |
| Add Subject  (see further testing below \*1) | 1 | Ask for new Details of  Subject | Yes |
| Modify Subject  (see further testing below \*2) | 2 | Displays Subjects list  with CRNs and Subject  names.  Asks for new Subject  Details | Yes |
| Display Subjects with all the details | 3 | Display Subjects with all the  details | Yes |
| Display Subjects with CRNS | 4 | Display Subjects with CRNS | yes |
| Display Lecturers with phone  numbers | 5 | Display Lecturers with phone  numbers | yes |
| Exit program | 6 | Program Exits | Yes |
| 1. **Add a Subject to List test**   When the user inputs option 1. The application first asks for the name of Subject, which could be any value. Then  The application displays the Select Lecturer Menu for selecting a Lecturer of Subject. | | | |
| **Select Lecturer Menu:**  Displays a menu for the user to select the Lecturer from the list of existing Lecturers  Also gives the user an option to Add a new Lecturer to the list if the required Lecturer is not present in the list,  this is always the last option in the menu.  e.g.  Select Lecturer for this subject:  0.Joe Test  1.Bill Gates  2.Steve Jobs  3.Add a new Lecturer  Testing input validity below: | | | |
| Out of range value      Add new Lecturer    Select Existing Lecturer  Character input | 100  3(in test case)  0  a | Error Message  Re-asks for input  Asks for Lecturer details  Selects the lecturer at option  Joe Test(in this case)  Error message,  Re-asks for input | Yes  Yes  Yes  Yes  Yes  Yes |
| Start time input(hh:mm) | -12:-12  we35  12,34  12:00 | sets time = 12:12 PM  error message, re-ask for  input  error message, re-ask for  input  Sets start time = 12:00PM | Yes  Yes  yes  Yes |
| Duration(hh:mm) | 3.5  3:00 | error message, re-ask for  input  Sets duration = 3Hrs | Yes  Yes |
| Subject Details | Name: Java  Lecturer:  David Warner  0890878  1 Regency R,  Regency Park,  SA,5010  Room: G29  Start time: 13:30  Duration: 3:00 | Subject Added to list  with the  correct details | Yes |
| **2.Modify Subject test**  Displays the List of Subjects with their CRNS and asks for user input. | | | |
| CRN value out of range | 0 | Displays Subject not Found | Yes |
| CRN Value as characters | ghkjl | Displays Error Message  and ask for CRN number  again | Yes |

Per above tests, the application functions as required.

# Fixed Errors.

* Implemented input validity in Select Lecturer Menu:

*Changed program to reject negative integer values*

* Input validity of Post Code when creating new Lecturer:

*Changed program to use Static method of isValidPostCode of Address Class to check if post code entered by user is valid or nor. If not valid user had to enter the post code again*