|  |
| --- |
| **North West Regional College**  **School of Science, Technology & Creative Industries**  Foundation Degree Science - Software Development  **Assignment Cover Sheet** |

|  |  |
| --- | --- |
| **Module Code:** | **COM476** |
| **Module Name:** | **Visual Programming & Data Structures** |
| **Assignment No:** | **1** |
| **Assignment Title:** | **Linked Lists** |
| **Lecturer(s):** | Peter Wisener |
| Moderator: | **Teresa Deeney** |
| **Moderation Date:** | **11/10/19** |
| **Release Date:** | **15/10/19** |
| **Submission Date:** | **12/11/19** |

|  |  |
| --- | --- |
| **Learning Outcomes Covered by Assignment** | K1, K2, I1, I2, P1, P2, T1, T3, T6 |

|  |  |
| --- | --- |
| **Definition of Plagiarism:** | Plagiarism is the act of taking or copying someone else’s work, including another student’s, and presenting it as if it were one’s own. Plagiarism is said to occur when ideas, texts, theories, data created artistic artifacts or other material are presented without acknowledgement so that the person considering the work is given the impression that what they have before them is the student’s own work when it is not. Plagiarism also occurs when a student’s own work is re-presented without being properly referenced. Plagiarism is a form of cheating and is a disciplinary offence. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Declaration:** | I declare that this is my own work and that any material I have referred to has been accurately and consistently referenced. I have read and understand the definition of plagiarism given above. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions. A mark of zero may be awarded and the reason for that mark will be recorded on my file. | | |
| **Student Name:** | **Sean Carlin** | | |
| **Student Signature:** |  | | |
| **Date Submitted:** | **12/11/19** | | |
| **Mark:** | **73** | | |
| **Assessor:** | **Peter Wisener** | **Date:** | **20/11/19** |

|  |  |  |  |
| --- | --- | --- | --- |
| **For Internal**  **Moderation Purposes Only** | **Sampling Date** | **Moderator** | **Signature** |
|  |  |

**Foundation Degree Software Development**

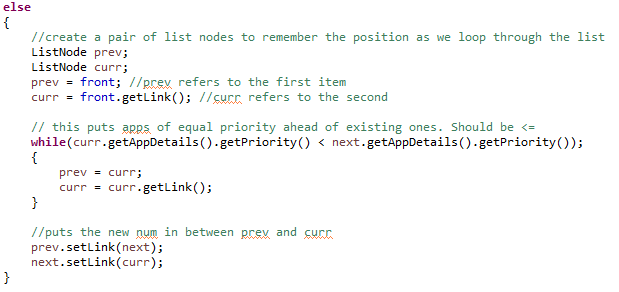
**Module: Visual Programming + Data Structures Assignment 1**

**Marking Scheme**

|  |  |
| --- | --- |
| **Interface Design** | **15** |
| Really well laid out form with nice touches such as images, icons and good use of panels. Reasonably user friendly although try to keep clicking to a minimum. Eg to edit an app you click Edit Applicant, click Find, click Yes to confirm, click Edit, click Update. This could actually be reduced to click Find, click Update. | **12** |
| **Use of Layout Managers** | **20** |
| Good use of GridBag including use of panels and development of the addComp method to pass in containers. Could have used weight x and y more effectively as reducing the size of the window means some components disappear completely. *See Appendix 3* | **17** |
| **Use of Data Structures** | **20** |
| Suitable list node but this code is also duplicated in the actual application queue class and shouldn’t be there. Add works to a point but not accurate for applications that go in the middle. *See Appendix 1.* DegreeAp class has a getPriority method but also an attribute and set method for priority. There is no need to store priority as it can be calculated using UCAS points. Should have encapsulated a method to allocate places in the queue class. | **12** |
| **Event Driven Coding and Documentation** | **25** |
| Grading calculator works well. However it stores the last grade selected in the combo box but the points is 0 so you have to change to a different grade and then change back to add the same grade twice. Add works well. Display is also good although when one application is added it shows the priority as the UCAS points. This is down to having an attribute for priority instead of using the method to calculate priority. *See Appendix 2*  Better use could be made of comments especially when names of variables, methods and classes could be more descriptive and accurate. Some unused code hasn’t been deleted.  Edit works ok, although a bit cumbersome as mentioned above. UCAS points haven’t been reset so Also when you move from Edit mode to Add mode the Edit Qualifications button doesn’t change back to Add qualifications. Delete is good.  Allocate places works fine in terms of displaying those who should get a place but it doesn’t actually allocate them a place in terms of changing the placeConfirmed attribute in the application objects. | **17** |
| **Validation** | **10** |
| Good use of custom exception to encapsulate validation within the Degree App class. Suitable messages used and invalid applications are not added. Only problem is that number format exceptions are not handled. (telephone number) | **9** |
| **Testing Documentation** | **10** |
| When testing validation rules you should test a range of values especially around the boundaries and make it clear what values were tested. Some good testing, although the several problems I found weren’t detected so the testing wasn’t effective. | **6** |
| **Total** | **100** |

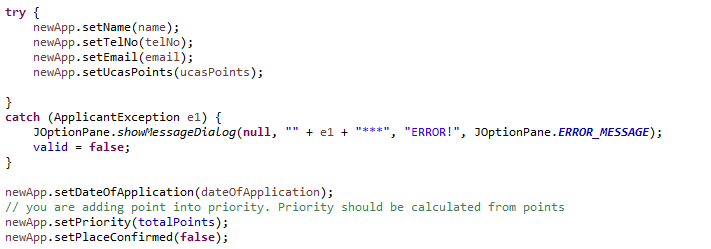
|  |  |
| --- | --- |
| **Final Mark** | 73 |

Appendix 1



In add method if a new application is the same priority level as an existing one it should go after it not before it so needs to be < =

Appendix 2



Appendix 3

