**LO3: design and develop an object oriented program to meet requirements.**

*3.1 Design an object oriented program.*

*3.2 Develop an object oriented program.*

Create a design document for a simple object-oriented solution from one of the example scenarios listed in the appendix, or one of your own of equal or greater complexity. Your solution must use **at least** three classes with data being passed between at least two of them.

Your design document should include:

* A brief statement of requirements
* Class diagram showing:
* Dependencies
* Inheritance where appropriate
* Attributes and methods
* Scope of attributes and methods
* Showing aggregation/association where appropriate
* Polymorphism where relevant
* Use of any pre-defined classes
* Sources of pre-defined classes
* Constructors for your classes
* Descriptions of the relationships between objects
* Definitions for message passing

You will then show, with the use of code snippets and screenshots, the development of your solution. Your notes should include:

* Use of a development environment
* Use of debugging tools
* Examples of data validation
* Error handling/reporting
* NOTE: it is useful to review your progress in your notes from time to time as this will help in task 4.
* Use of appropriate syntax
* Variables and data structures
* Selection statements (If – else, CASE/SWITCH etc.)
* Iterations (Loops – while, repeat)
* Programming standards: e.g. use of comments, code layout, indentation.

**Task 4**

**Test and document an object oriented program.**

*4.1 Test an object oriented program.*

*4.2 Document appropriate action to correct errors.*

*4.3 Create technical documentation for the support and maintenance of the program.*

Now you have completed your program, you need to document the testing of it against your requirements.

* Develop a test plan to test your program’s functionality
* test, date, expected result, actual result, corrective action

Then work through your test plan and record the actual results, noting:

* Screenshots any error messages as part of your testing
* Show use of debugging tools as part of your testing

Review your software solution against specifications requirements

You can also include any interim reviews you carried out in task 3.

* What went well?
* What did not go so well and why?
* What are the future refinements you would make if you had time?

Create a short use guide to your software solution.

* How to run the software
* Operating instructions