



Unit:
**Designing and Developing Object-Oriented
Computer Programs**

Assignment title:
Home Appliance Rental System

Sample Assignment

Marking Scheme

Markers are advised that, unless a task specifies that an answer be provided in a particular form, then an answer that is correct (factually or in practical terms) **must** be given the available marks. If there is doubt as to the correctness of an answer, the relevant NCC Education materials should be the first authority.

This marking scheme has been prepared as a **guide only** to markers and there will frequently be many alternative responses which will provide a valid answer.

Each candidate's script must be fully annotated with the marker's comments (where applicable) and the marks allocated for each part of the tasks.

It is essential that this marking scheme is applied. Marks will not be accepted if is not.

Throughout the marking, please credit any valid alternative point.

Where markers award half marks in any part of a task, they should ensure that the total mark recorded for the task is rounded up to a whole mark.

Marker's comments:

Moderator's comments:

Mark:

Moderated mark:

Final mark:

Penalties applied for academic malpractice:

Task 1 (LO1, 2, 3) – 60 Marks

- a) You need to design and implement a program (written in C#) which fulfils all the requirements as outlined above.

(50 marks)

Mark Scheme

0-14 marks	15-19 marks	20-29 marks	30-34 marks	35-50 marks
<p>Program has ONE (1) functional requirements or has TWO (2) functional requirements with many major errors.</p> <p>Little to no evidence that considerations have been made for customer interaction, and little or no attempt to handle invalid user inputs.</p> <p>No/Inappropriate GUI controls.</p>	<p>Program has TWO (2) functional requirements with some major errors or has THREE (3) functional requirements with some major errors.</p> <p>Insufficient evidence that considerations have been made for all types of customer interaction and attempt to handle a small proportion of invalid user inputs.</p> <p>Very few appropriate GUI controls and poorly laid out.</p>	<p>Program has TWO (2) functional requirements with some minor errors or THREE (3) functional requirements with some major errors.</p> <p>Sufficient evidence that considerations have been made for customer interaction and some invalid user inputs handled.</p> <p>Some appropriate controls are used and neatly laid out.</p>	<p>Program has THREE (3) functional requirements with some minor errors or FOUR (4) functional requirements with a few major errors.</p> <p>Good evidence that considerations have been made for customer interaction and majority of invalid user inputs handled consistently.</p> <p>Good use appropriate GUI controls and neatly laid out.</p>	<p>Program satisfies FIVE (5) functional requirements with a few minor errors or any FOUR (4) functional requirements without errors.</p> <p>Strong evidence that considerations for customer interaction have been made and coherently handling invalid user inputs.</p> <p>Excellent use appropriate GUI controls consistently and neatly laid out.</p>

- b) The quality of your program will be assessed in terms of program structure, OOP principles including encapsulation, algorithms using appropriate control structures (loops and selections), and readability including appropriate comments

(10 marks)

Mark Scheme

0-2 marks	3 marks	4-5 marks	6 marks	7-10 marks
<i>Incorrect algorithms, not using selections and loops.</i>	<i>Inefficient algorithms, not using selections and loops appropriately.</i>	<i>Algorithms use some selections and loops appropriately.</i>	<i>Efficient algorithms, properly using selections and loops.</i>	<i>Very efficient algorithms, with excellent use selections and loops.</i>
<i>Objects are not self-contained and private data is not hidden.</i>	<i>Very small proportion of objects are self-contained with private data hidden.</i>	<i>Some objects are self-contained with private data hidden.</i>	<i>Most objects are self-contained with private data hidden.</i>	<i>All objects are self-contained with private data hidden.</i>
<i>No or inappropriate naming conventions.</i>	<i>Occasional but insufficient use of naming conventions but may be inappropriate.</i>	<i>Some variables/ classes show some use of appropriate naming conventions.</i>	<i>Variables/ classes mostly use appropriate naming conventions.</i>	<i>Variables/ classes all use appropriate/meaningful naming conventions.</i>
<i>No or little comments.</i>	<i>Inadequate comments.</i>	<i>Adequate comments added.</i>	<i>Appropriate comments added.</i>	<i>Appropriate comments added consistently throughout the program.</i>

Task 2 (LO1, 4) – 20 Marks

You will need to develop a test plan and implement it. You should write a FIVE HUNDRED (500) word report about the plan, which should include the test plan, purpose of each test, together with test scripts*, and the testing results* with full details.

An additional FIVE HUNDRED (500) word discussion should be included to justify how the data was selected and executed.

*There is no word limit for the test scripts and testing results.

The minimum requirements for a pass mark include developing and implementing a test plan including test cases with some possible/invalid inputs for any TWO (2) functions mentioned above to ensure the functions handle the inputs as they should.

Possible exceptions should be considered and tested to make sure they are properly handled.

Mark Scheme

0-5 marks	6-7 marks	8-11 marks	12-13 marks	14-20 marks
No/unrecognisable test plan.	Test plan poorly formatted with major omissions (scripts/results).	Test plan contains some appropriate tests/formatting/scripts/results.	Detailed appropriate test plan/results/scripts with some weaknesses.	Comprehensive and detailed test plan/clearly linked to results/scripts.
No or vague discussion of how data selected/executed.	Haphazard discussion of data selection/execution of tests.	Some valid discussion of how data selected/executed.	Detailed and valid discussion of how most data was selected/executed.	Clear, concise systematic discussion of selection of all data.
No/Few exceptions are caught and handled.	A few exceptions missing.	Exceptions are caught and handled.	Exceptions are caught and handled.	Exceptions are caught and handled appropriately.
Test cases cover no/little range of possible inputs and few invalid inputs.	Test cases cover an extremely limited range of possible inputs and few invalid inputs.	Test cases cover some possible inputs and some invalid inputs.	Test cases cover appropriate range of possible inputs and invalid inputs. Some minor weaknesses.	Test cases span appropriate range of possible inputs and invalid inputs.
No or little use of the	Inadequate use of the	Use of some of the	Systematic use of most	Systematic use of all

<i>available testing strategies delivered within the module.</i>	<i>available testing strategies delivered within the module.</i>	<i>available testing strategies delivered within the module.</i>	<i>testing strategies delivered within the module.</i>	<i>available testing strategies delivered within the module.</i>
--	--	--	--	--

Task 3 (LO5) – 20 Marks

You need to:

- Create a fully detailed class diagram, which should be accompanied by a FIVE HUNDRED (500) word justification of the design of your classes. Note that you are **not** allowed to use any tools that automatically build class diagrams.

Mark Scheme

0-5 marks	6-7 marks	8-11 marks	12-13 marks	14-20 marks
<i>Class members and relationships have no/little description.</i>	<i>Class members and relationships are described inadequately.</i>	<i>Some class members and relationships have some valid description / justification.</i>	<i>Most class members and relationships are well described and satisfactorily justified.</i>	<i>All class members and relationships are well described consistently and highly justified.</i>
<i>No classes and relationships in the class diagram incorrectly represent the program.</i>	<i>Very few classes and relationships in the class diagram correctly represent the program.</i>	<i>Some classes and relationships in the class diagram correctly represent the program.</i>	<i>Most classes and relationships in the class diagram correctly represent the program.</i>	<i>All classes and relationships in the class diagram correctly represent the program.</i>
<i>No notation is correct.</i>	<i>Very little notation is mostly correct.</i>	<i>Some of the notation is correctly used.</i>	<i>Most of the notation is correctly used.</i>	<i>Correct use of notation consistently used.</i>
<i>No/Few class structures and relationships are illustrated correctly.</i>	<i>A few class structures and relationships are illustrated correctly.</i>	<i>Many class members and relationships are illustrated correctly.</i>	<i>All class members and relationships are correctly illustrated with minor errors.</i>	<i>Class members and relationships are all illustrated correctly.</i>

Learning Outcomes matrix

Task	Learning Outcomes assessed	Marker can differentiate between varying levels of achievement
1	1, 2, 3	Yes
2	1, 4	Yes
3	5	Yes

Grade descriptors

Learning Outcome	Fail	Referral	Pass	Merit	Distinction
Design object-oriented programmes to address loosely defined problems	Demonstrates little to no ability to adequately use design principles to effectively create an artefact to solve an identified issue.	Demonstrates an extremely limited ability to adequately use design principles to effectively create an artefact to solve an identified issue.	Demonstrates a satisfactory ability to use design principles to effectively create an artefact to solve an identified issue.	Demonstrates a very good ability to use design principles to effectively create an artefact to solve an identified issue.	Demonstrates an excellent ability to use design principles to effectively create an artefact to solve an identified issue.
Implement object-oriented programmes from well-defined specifications	Demonstrates little to no ability to adequately implement an artefact to solve an identified issue.	Demonstrates an extremely limited ability to adequately implement an artefact to solve an identified issue.	Demonstrates a satisfactory ability to implement an artefact to solve an identified issue.	Demonstrates a very good ability to implement an artefact to solve an identified issue.	Demonstrates an excellent ability to implement an artefact to solve an identified issue.
Develop object-oriented programmes that reflect established programming and software engineering practice	Demonstrates little to no ability to adequately develop an artefact to solve an identified issue.	Demonstrates an extremely limited ability to adequately develop an artefact to solve an identified issue.	Demonstrates a satisfactory ability to develop an artefact to solve an identified issue.	Demonstrates a very good ability to develop an artefact to solve an identified issue.	Demonstrates an excellent ability to develop an artefact to solve an identified issue.
Develop test strategies and apply these to object-oriented programmes	Demonstrates little to no ability to adequately review the effectiveness and appropriateness of information, data and results by failing to use pre-defined techniques and/or criteria.	Demonstrates an extremely limited ability to adequately review the effectiveness and appropriateness of information, data and results due to insufficient use of pre-defined techniques and/or criteria.	Demonstrates a satisfactory ability to review the effectiveness and appropriateness of information, data and results due to adequate use of pre-defined techniques and/or criteria.	Demonstrates a very good ability to review the effectiveness and appropriateness of information, data and results that goes beyond the minimum required to pass due to an accurate use of pre-defined techniques and/or criteria.	Demonstrates an excellent ability to comprehensively review the effectiveness and appropriateness of information, data and results due to a meticulous use of pre-defined techniques and/or criteria.

Develop design documentation for use in program maintenance and end-user documentation	Demonstrates little to no ability to adequately develop an artefact to solve an identified issue.	Demonstrates an extremely limited ability to adequately develop an artefact to solve an identified issue.	Demonstrates a satisfactory ability to develop an artefact to solve an identified issue.	Demonstrates a very good ability to develop an artefact to solve an identified issue.	Demonstrates an excellent ability to develop an artefact to solve an identified issue.
---	---	---	--	---	--