TAYLOR'S UNIVERSITY

Wisdom • Integrity • Excellence

# FINAL ASSESSMENT
# AUGUST 2021 SEMESTER

| | |
|---|---|
| **MODULE NAME** | **: ADVANCED PROGRAMMING** |
| **MODULE CODE** | **: ITS66704** |
| **EXAM DURATION** | **: 24 HOURS** |
| **DATE** | **: 13/12/21 08:00AM** |
| **SUBMISSION DEADLINE** | **: 14/12/21 08:00AM (M'SIA TIME GMT+8)** |

**This paper consists of EIGHT (8) printed pages, inclusive of this page.**

**Instruction to Candidates:**

1. Answer ALL questions.
2. This is an open book examination. Student is not allowed to transcribe directly (copy and paste) any material from another source into their submission.
3. The Turnitin similarity for this module is 20% overall and lesser than 1% from a single source excluding program source codes.
4. Severe disciplinary action will be taken against those caught violating assessment rules such as colluding, plagiarizing or transcribing.
5. The final assessment answers handed in should be with a spacing of 1.5 and a font of 12pt Times New Roman.
6. Name your answer file as ITS66704_XXXXXX_FinalAssessment.pdf where XXXXXX is your Student ID. Then, upload to TiMES portal via the link "Final Assessment submission" in module page. (Do not submit the question paper)
7. The breakdown of exam questions by Module Learning Outcome(s) and its associate weightage is as follows:

| MLO | Section(s)/ Question(s) | Marks |
|---|---|---|
| MLO2 | Section A: Question 1 & 2 | / 50 |
| MLO1 | Section B: Question 1, 2, 3 & 4 | / 50 |
| | TOTAL | / 100 |

8. Start each answer on a separate page.
9. Students are not allowed to consult with any party with regard to any part of the questions throughout the assessment period.

## Objectives / Module Learning Outcomes

The objectives of this special assessment is to enable the students to:

1. Describe the concepts of advanced object-oriented topics including Exception Handling, IO Streams, Generics, Collection Framework, Event Handling and GUI Programming.
2. Apply problem solving skills to evaluate and solve specific topics in advanced object-oriented problem and programs.

## Section A – Short Program (50 marks) MLO 2

1. Create a Java program that demonstrates the application of the fundamental concepts of object-oriented programming (OOP), and advanced programming concepts.

   (40 marks)

   An international car dealership has decided to convert their sales system to a fully Object Oriented (OO) System using Java programming languages. Currently, the dealership has two types of Car; family car and sport car. Both types have attributes Model and Price. Family cars have free service value but no discount while sport cars have discount but no free service value.

   *Example of data for family cars*

   | | | | |
   |---|---|---|---|
   | Family Car 1 | Model = BMW320 | Price = $200,000.00 | Free Service = $5,000.00 |
   | Family Car 2 | Model = BMW525 | Price = $300,000.00 | Free Service = $9,000.00 |

   *Example of data for sport cars*

   | | | | |
   |---|---|---|---|
   | Sport Car 1 | Model = BMW M3 | Price = $400,000.00 | Discount = 10% |
   | Sport Car 2 | Model = BMW 850 | Price = $600,000.00 | Discount = 15% |

   To test the validity of the OOP structures, few transactions such as purchase, increase price, decrease price should be conducted in the overall solution.

   **Output Examples**

   Output screen example

```
***** FAMILY CAR *****
Model         : BMW320
Price         : $200000.0
Free Service : $6000.0

***** FAMILY CAR *****
Model         : BMW525
Price         : $300000.0
Free Service : $9000.0

***** SPORT CAR *****
Model         : BMWM3
Price         : $400000.0
Discount      : 10%
```

Sample transactions output screen example

```
Increase $5000 to BMW320
Decrease $400000 from BMW525 - Rejected
Decrease $10000 from BMW525
Increase $10000 to BMWM3
Purchase BMW320 – price $205000
Purchase BMW850 – price $510000

***** FAMILY CAR *****
Model         : BMW320
Price         : $205000.0
Free Service : $6000.0

***** FAMILY CAR *****
Model         : BMW525
Price         : $290000.0
Free Service : $9000.0

***** SPORT CAR *****
Model         : BMWM3
Price         : $410000.0
Discount      : 10%

***** SPORT CAR *****
Model         : BMWM850
Price         : $600000.0
Discount      : 15%
```

Your program must demonstrate/contain the followings concepts/keywords/tasks:

a) Instantiation – The main program should instantiate of the four (4) objects above, and an instantiation of 1 (one) object from an additional subclass. (6 marks)

b) Encapsulation – All attributes must be fully encapsulated. (2 marks)

c) Inheritance – Provide one (1) superclass, and three (3) subclasses. Each class must have at least two (2) attributes. Do add necessary subclass and attributes to fulfill these requirements. (9 marks)

d) Polymorphism – There should be one (1) instance of method overloading. (2 marks)

e) Abstract – There should be one (1) abstract method implemented. (3 marks)

f) Based on the program, draw a full UML class diagram. (4 marks)

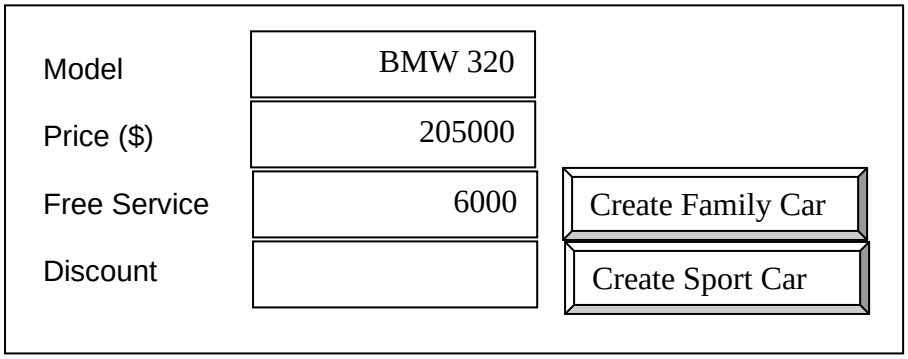
g) Write a short report to provide brief description of your additional subclass and attributes, and to point out where all the above concepts are used in your program. (6 marks)

h) Include the proper source code of your program, and the output screen shots in the report. (8 marks)

(TOTAL 40 marks)

2. Convert your program to full Java Swing or FX GUI program (see Figure below) to enable instantiation by inputting data into a text field controls. You do not need to convert the output into a GUI based controls.

GUI output screen example




(10 marks)

*See the marking scheme for full marks allocations*

## Section B – Short Questions (50 marks) MLO 1

Answer **ALL** questions in the answer report, starting each question in a new page.

1. Security and Generics (20 marks)

   (a) Provide Java code segment to demonstrate the use of an **interface** that contains two (2) methods. In addition, discuss the advantage and disadvantage of an interface method in comparison to an abstract method. (10 marks)

   (b) Provide Java code segment to demonstrate the ability of generic to simplify method overloading. (5 marks)

   (c) Discuss how final prefix can be used to create a secure class. (5 marks)

2. The implementation of certain advanced OO concepts needs to be incorporated early in the design stage. This is to avoid costly modification in later stages. (10 marks)

   (a) Discuss how "IS-A" rule assists us in building accurate UML class diagram in design stage. (3 marks)

   (b) Briefly explain how Java overcomes the absence of **multiple inheritance** support. (3 marks)

   (c) Discuss the difference and similarity of method **overloading** and method **overriding**. (4 marks)

3. Exceptions Handling (10 marks)

   The use of exception handling in Java can be both voluntary and involuntary (compulsory). Create a simple Java program to demonstrate the use both methods, and discuss the reason for using Java exception handling voluntarily. (10 marks)

4. Serialized Data Reading (10 marks)

   Provide a simple Java program to show how an object of any three attributes (one String, one double, and one integer) is read from a serialized data file. Include a discussion on how the program is able to read the data at a precise location in the data file. (10 marks)

*See the marking scheme for full marks allocations.*

**END OF QUESTION SECTION**

# Marking Schemes & Rubrics

## SECTION A

| Question 1 (a) - Short Program (40 marks) | marks |
|---|---|
| **Program** | |
| Demonstrate Instantiation of 4 objects from 2 standard subclasses | 4 |
| Demonstrate Instantiation of own object from additional subclass | 2 |
| Demonstrate Encapsulation | 2 |
| Demonstrate Inheritance of 1 superclass and 2 subclasses with proper attributes | 5 |
| Demonstrate Inheritance of additional subclass with proper attributes | 4 |
| Demonstrate Polymorphism - method overloading | 2 |
| Demonstrate Abstract - 1 implemented abstract | 3 |
| **Documentation** | |
| Brief description of the program | 1 |
| Documentation of concepts | 5 |
| UML Class Diagram | 3 |
| Source code | 2 |
| Program output (Screen shots) | 3 |
| Output closely resembles the sample outputs | 2 |
| Overall Program Style (Aesthetics, comment, naming, indentation) | 2 |
| **TOTAL Q1** | **40** |

| Question 2 – UI Input Controls (10 marks) | marks |
|---|---|
| Provide proper Java FX/Swing Model View setup using standard IDE | 2 |
| Demonstrate proper Label and Textfield for objects instantiation | 3 |
| Demonstrate proper Button trigger event | 2 |
| Demonstrate successful instantiation of the objects | 3 |
| **TOTAL Q2** | **10** |

| | |
|---|---|
| **TOTAL SECTION A** | **50** |

## SECTION B

| Question 1 (a) - Interface (10 marks) | marks |
|---|---|
| Java code of interface with two methods | 3 |
| Interface implementation | 3 |
| Discuss advantage and disadvantage of an interface | 4 |
| **TOTAL Q1 (b)** | **10** |

| Question 1 (b) - Generics (5 marks) | marks |
|---|---|
| Generics Java Code example | 3 |
| Simplification of method overloading | 2 |
| **TOTAL Q1 (b)** | **5** |

| **Question 1 (c) - Final usage (5 marks)** | **marks** |
|---|---|
| How final attribute contributes to security | 2 |
| How final method contributes to security | 2 |
| How final class contributes to security | 1 |
| **TOTAL Q1 (c)** | **5** |

| **Question 2 (a) – IS-A Rule (3 marks)** | **marks** |
|---|---|
| IS-A rule explanation | 1 |
| Use of IS-A rule in UML class diagram | 2 |
| **TOTAL Q2 (a)** | **3** |

| **Question 2 (b) – Multiple Inheritance (3 marks)** | **marks** |
|---|---|
| Workaround solution | 1 |
| Implementation example | 2 |
| **TOTAL Q2 (b)** | **3** |

| **Question 2 (c) – Overriding and Overloading polymorphism (4 marks)** | **marks** |
|---|---|
| Similarity of Overloading and Overriding | 3 |
| Difference of Overloading and Overriding | 3 |
| **TOTAL Q2 (c)** | **4** |

| **Question 3 - Exception handling (10 marks)** | **marks** |
|---|---|
| Proper example of voluntary and compulsory exceptions | 4 |
| Implementation of Try, Catch and Finally block | 3 |
| Reasons for using voluntary exception handling | 3 |
| **TOTAL Q3** | **10** |

| **Question 4 – Serialized data reading (10 marks)** | **marks** |
|---|---|
| Proper Java example of serialized data reading | 4 |
| Implementation of String, integer and double | 3 |
| Discussion of how data is read | 3 |
| **TOTAL Q4** | **10** |

| | |
|---|---|
| **TOTAL SECTION B** | **50** |

| | |
|---|---|
| **GRAND TOTAL (SECTION A + SECTION B)** | **100 marks** |

## MARKING RUBRICS

**For EACH criterion of marks allocated, the following rubrics will be applied:**

| 100% of allocated marks | 75% of allocated marks | 50% of allocated marks | 25% allocated marks | 0% allocated marks |
|---|---|---|---|---|
| • Complete understanding of the problem<br>• A plan that could lead to a correct solution with no algorithmic errors<br>• Correct solution | • Misinterprets minor part of the problem<br>• Substantially correct solution with minor omission or procedural error | • Misinterprets major part of the problem<br>• Partially correct solution but with major fault<br>• Computational error, partial solution for problem. | • Completely misinterprets the problem<br>• Substantially inappropriate solution | • No attempt<br>• No answer or wrong answer based upon an inappropriate solution |

**END OF MARKING SCHEME SECTION**