Module Information Summary

Section 1
Module Description - General

Module Code	**ITS66704**
Module Name	**Advanced Programming**
Year Level (Module)	**2**
Credit Value	**4**
Semester(s) Offered	**March** **August**
Module Leader	**Steve Teoh Chee Hooi (Steve.Teoh@taylors.edu.my)**

Synopsis

This module emphasises on implementing advanced object-oriented principles using Java. This module comprises of five major advanced topics including exception handling, file programming, event handling, graphical user interface programming, collections framework and generics. Students are first introduced to the concept of error trapping, the need and consequences of not implementing error trapping in development. Next, students will learn the types of exception handling in object-oriented programming and its implementation in Java. Students are then exposed to, the concept of file programming and methods of reading and writing to files and external resources using the appropriate streams. Event handling and GUI programming constitutes the major portion of this subject and is heavily assessed in the group assignment. These two significant topics will give students a detailed hands-on experience on how to develop a GUI based application and implement event handling using Java. Moving on students will then be introduced to the Collections framework where they are exposed to the implementation of data structures in Java using the Collections library. Students are taught how to use commonly used structures such as List, Set and Maps and the merits and demerits of each data structures. Generics is the final topic in this module to teach students on how to create template programs which promote reusability concept in programming. This module will be assessed via a practical test, group assignment and final examination. Information and knowledge will be managed and transferred using both traditional and digital methods through approaches such as lecture, discussion, demonstration (practical session), case studies and projects through active, collaborative and multidisciplinary learning approaches. The practical test will serve as a midpoint check which will access the students understanding on the first two topics in this module. Group assignment assessment carries the most substantial weight in this module and is assessed in two parts namely program development and demonstration of the working program. With this component, they will collaborate with students from another school/faculty to strengthen their projects as part of multidisciplinary learning. The last evaluation for this module is 2 hours written final exam which will test the overall understanding of the concepts and implementation covered in this module. This module applies two different teaching and learning pedagogies, (1) Guided Learning and (2) Problem-Based Learning. The Guided Learning allows the facilitator to play a proactive role in terms of teaching, and motivate and guiding the students to their advanced concepts of programming. When the advanced skill of programming is learnt, then Problem-Based Learning pedagogy will engage the students in problem-solving activities. Delivery of these two pedagogies will be via the practical computer laboratory sessions, take home problem sets, and minimal face-to-face lectures where necessary to increase independence learning. The learning is facilitated mostly through lecture, practical computer laboratory and blended learning. In other words, for each topic, students will get to learn the concept as well implement in two different avenues. During the lecture session, after the introduction of each concept, a small program or fragment of code will be executed in class for students to understand the implementation of the concept. Exercises and discussion in the lab sessions give exposure to code tracing and implementation. With the motive “assessment for learning”, the assessments are spread as 70% in course assessment and 30% final examination with a heavy concentration on problem solving in programming logics and advanced of Object-Oriented concepts. with a heavy concentration on problem solving in programming logics and advanced level of Object-Oriented Concepts. The in-course assessments involve assignments working as group and practical test to evaluate the object-oriented knowledge.

General Pre-Requisite	**ITS63304 Object Oriented Programming**
Module Pre-requisite	**Object Oriented Programming**
General Co-requisite	-
Module Co-requisite	-
General Anti-requisite	-
Module Anti-requisite	-

Module Owner	**School of Computer Science**
Domain Name (for free electives only)	**Science, Technology and Society**
Module Offered as	**Primary Major - Core**

Section 2
Module Description - Module Learning Outcomes (MLO)

No./ID	Description	MLO %
MLO1	Describe the concepts of advanced object-oriented topics including Exception Handling, IO Streams, Generics, Collection Framework, Event Handling and GUI Programming.	10
MLO2	Apply problem solving skills to evaluate and solve specific topics in advanced object-oriented problem and programs.	50
MLO3	Demonstrate capability to interact positively within an interdisciplinary peer group, consider other view-points, and foster stable and harmonious relationships in solving computational problems related to object-oriented language.	10
MLO4	Present the outcome of the program developed using object-oriented language with the appropriate integrated environment.	30
		100

Section 3
Module Description - Align MLO to PLO & TGC Sub-Attribute

No	Module Learning Outcomes (MLO)	MLO %	TGC Attributes	TGC Sub-Attributes	Description of sub-attributes
1	MLO1	10	TGC 1 Discipline-specific knowledge and Skills	1.1	Knowledge - Demonstrate a broad and coherent theoretical and technical knowledge to communicate and understanding relating to the discipline-specific content
				1.2	Comprehension - Demonstrate comprehension of disciplinary concepts
2	MLO2	50	TGC 2 Problem Solving, Critical and Creative Thinking Skills	2a.1	Identify the problem
				2a.2	Propose solutions to existing and emerging problems
				2a.3	Implement a solution
				2b.2	Creative thinking
3	MLO3	10	TGC 6 Social Competencies	6.1	Take perspective and empathize with others (Measuring: Empathy)
4	MLO4	30	TGC 5 Personal Competencies	5.3	Persevere in overcoming obstacles and setbacks (Measuring: Grit/Resilience)
				5.5	Set and monitor progress towards personal and academic goals (Measuring: students goal-setting ability i.e set, monitor and fulfill goals over set timeline)

Section 4
Module Description - Transferrable Skills

Transferable Skills

TGC	TGC Attribute	Description of TGC Attributes	Sub-Attribute	Description of sub-attributes
TGC 1	Discipline-specific knowledge and Skills	Discipline-specific knowledge and skills refers to the ability to demonstrate professional competence, adapt discipline specific knowledge, and be able to integrate knowledge across different perspectives.	1.1	Knowledge - Demonstrate a broad and coherent theoretical and technical knowledge to communicate and understanding relating to the discipline-specific content
			1.2	Comprehension - Demonstrate comprehension of disciplinary concepts
TGC 2	Problem Solving, Critical and Creative Thinking Skills	Problem Solving, Critical and Creative Thinking Skills refer to the ability to rationally, critically and creatively analyze, synthesize and evaluate evidence to arrive at a solution or conclusion.	2a.1	Identify the problem
			2a.2	Propose solutions to existing and emerging problems
			2a.3	Implement a solution
			2b.2	Creative thinking
TGC 6	Social Competencies	Social Competencies refer to the ability to empathize with others, interact positively with them and foster stable and harmonious relationships.	6.1	Take perspective and empathize with others (Measuring: Empathy)
TGC 5	Personal Competencies	Personal Competencies refer to the ability to be self-aware and to self-regulate emotions through skillful management of one's personal goals, intentions, responses and behaviour.	5.3	Persevere in overcoming obstacles and setbacks (Measuring: Grit/Resilience)
			5.5	Set and monitor progress towards personal and academic goals (Measuring: students goal-setting ability i.e set, monitor and fulfill goals over set timeline)

Section 5
Module Description - Align MLO to Assessment Tasks and T&L Strategies

Description of Assessment Components

The assessment for this module is external (Taylor's College)	**NO**
Resit Opportunity	**TU-UWE**

Assessment Tasks

Assessment Task	Weight	MLO Assessed	TGC Assessed	Due Date	Maximum Mark (Task Level)	Maximum Mark (MLO Level)
Assessment Task 1: Practical Test	10%	MLO1	1	5	10	10
Assessment Task 2: Group Assignment	50%	MLO2	2	12	50	50
Assessment Task 3: Program Presentation	10%	MLO3	6	13	10	10
Final Examination	30%	MLO4	5	16	30	30
	100%					

Section 6
Module Description - Resit Assessment

Description of Resit Assessment

Resit Opportunity **TU-UWE**

Assessment Tasks

Assessment Task	Weight	MLO Assessed	TGC Assessed	Maximum Mark (Task Level)	Maximum Mark (MLO Level)
Resit Coursework	70%	MLO1 MLO2 MLO3	TGC 1 TGC 2 TGC 6	70	10 50 10
Resit Final Examination	30%	MLO4	TGC 5	30	30
	100%				

Section 7
Module Description - Rubrics for Assessment Tasks

<table>
<tr><th>Assessment Task</th><th>MLO Assessed</th></tr>
<tr><td>Assessment Task 1: Practical Test</td><td>MLO1

Describe the concepts of advanced object-oriented topics including Exception Handling, IO Streams, Generics, Collection Framework, Event Handling and GUI Programming.

<table>
<tr><th colspan="6">Assessment Task 1: Practical Test: (MLO Assessed: MLO1)</th></tr>
<tr><th>Criteria</th><th>Weightage</th><th>Beginning: 0 – 49%</th><th>Developing: 50 – 64%</th><th>Mastering: 65 – 79%</th><th>Outstanding: 80 – 100%</th></tr>
<tr><td>Demonstrate the skills to identify the problem Problem solving constructing a program logic and design using pseudo-code, algorithm and encoding it into a notation using sequential processing, selection for decision-making and iteration for repetitive control including: *Exception Handling *File Programming *Event Handling *User Interface Control *Collection Framework * IO Streams * Generics</td><td>10.00%</td><td>(0-4) No solution provided or critical elements of the solution are missing or significantly flawed. Solution does not demonstrate sufficient understanding of the problem and/or any reasonable directions to solve the problem.</td><td>Solution demonstrates a viable approach toward solving the problem but contains some major pitfalls, errors/flaws or limitations.</td><td>Mastering (7-8) Solution is mostly correct, satisfying most of the above criteria under the exemplary category, but contains some minor pitfalls, errors/flaws or limitations.</td><td>Outstanding (9-10) Solution presented solves the problem stated correctly and meets all requirements of the problem. Solution is clearly presented.Solution represents an elegant and effective way to solve the problem and is not overly complicated than is necessary.</td></tr>
</table>
</td></tr>
<tr><td>Assessment Task 2: Group Assignment</td><td>MLO2

Apply problem solving skills to evaluate and solve specific topics in advanced object-oriented problem and programs.

<table>
<tr><th colspan="6">Assessment Task 2: Group Assignment: (MLO Assessed: MLO2)</th></tr>
<tr><th>Criteria</th><th>Weightage</th><th>Beginning: 0 – 49%</th><th>Developing: 50 – 64%</th><th>Mastering: 65 – 79%</th><th>Outstanding: 80 – 100%</th></tr>
<tr><td>Present the outcome of the program developed using object-oriented language with the appropriate integrated environment.</td><td>50.00%</td><td>Can articulate, present and explain none or some of the solution and outcome of the application which aligns to the design and the chosen theme developed using object-oriented language with an appropriate development tool.</td><td>Can articulate, present and explain adequate/considerable solution and outcome of the application which aligns to the design and the chosen theme developed using object-oriented language with an appropriate development tool.</td><td>Can articulate, present and explain mostly the solution and outcome of the application which aligns to the design and the chosen theme developed using object-oriented language with an appropriate development tool.</td><td>Outstanding (5) Can articulate, present and explain the solution and outcome of the application which aligns to the design and the chosen theme developed using object-oriented language with an appropriate development tool.</td></tr>
</table>
</td></tr>
<tr><td>Assessment Task 3: Program Presentation</td><td>MLO3

Demonstrate capability to interact positively within an interdisciplinary peer group, consider other view-points, and foster stable and harmonious relationships in solving computational problems related to object-oriented language.

<table>
<tr><th colspan="6">Assessment Task 3: Program Presentation: (MLO Assessed: MLO3)</th></tr>
<tr><th>Criteria</th><th>Weightage</th><th>Beginning: 0 – 49%</th><th>Developing: 50 – 64%</th><th>Mastering: 65 – 79%</th><th>Outstanding: 80 – 100%</th></tr>
<tr><td>Organisation of the presentation, presentation format and style.</td><td>10.00%</td><td>Demonstrate little attention to and execution of a wide range of conventions particular to a the selected theme for the application development including organization, content, presentation formatting and style</td><td>Demonstrate adequate attention to and execution of a wide range of conventions particular to a the selected theme for the application development including organization, content, presentation formatting and style</td><td>Demonstrate competent attention to and execution of a wide range of conventions particular to a the selected theme for the application development including organization, content, presentation formatting and style</td><td>Demonstrate detailed attention to and successful execution of a wide range of conventions particular to a the selected theme for the application development including organization, content, presentation formatting and style</td></tr>
</table>
</td></tr>
<tr><td>Final Examination</td><td>MLO4

Present the outcome of the program developed using object-oriented language with the appropriate integrated environment.

<table>
<tr><th colspan="6">Final Examination: (MLO Assessed: MLO4)</th></tr>
<tr><th>Criteria</th><th>Weightage</th><th>Beginning: 0 – 49%</th><th>Developing: 50 – 64%</th><th>Mastering: 65 – 79%</th><th>Outstanding: 80 – 100%</th></tr>
<tr><td>Final Exam, as per the marks distribution</td><td>30.00%</td><td>(0-8)</td><td>(9-17)</td><td>(81-24)</td><td>(24-30)</td></tr>
</table>
</td></tr>
</table>

Section 8 Module Description - Hurdle Assessment

Hurdle Assessment Guideline

Hurdle assessment guideline for the module: A student must achieve at least 50% for the overall assessment and a final grade of C to pass the module. A student who obtains a minimum of 40% for the overall assessment and overall grade of D or higher for the module may be allowed to resit the examination. The maximum passing grade awarded for the resit examination will be a grade C. A student who obtains 39% and below for the final assessment, will result in failing the module irrespective of the overall marks earned, even though he/she has achieved 50% or more in the overall assessment. He/she will not be allowed to resubmit the final assessment

normal-long

	Physical F2F (Interactive Lecture)	Physical F2F (Tutorial)	Physical F2F (Practical)	Physical F2F (Other)	Online Synchronous F2F (Interactive Lecture)	Online Synchronous F2F (Tutorial)	Online Synchronous F2F (Practical)	Online Synchronous F2F (Other)	NF2F Independent Learning	Assessment Task (Physical F2F)	Assessment Task (Online Synchronous F2F)	Assessment Task Independent Learning for Assessment (Asynchronous)	Student Learning Time (SLT)
	HOUR												
Week - 1	1				2				3				6
	Introduction and Overview MLO1				Introduction and Overview MLO1				Introduction and Overview MLO1				6
Week - 2	1		2		2				4				9
	Exception Handling and Text IO MLO1		Implementing Exception Handling on a simple program MLO1		Exception Handling and simple file system MLO1				Exception Handling MLO1				9
Week - 3	1		2		2				4				9
	Exception Handling and Text IO MLO1		Create simple program to implement File Class MLO1		Exception Handling and simple file system MLO1				Exception Handling MLO1				9
Week - 4	1		2		2				4				9
	Basics User Interface MLO2		Basics User Interface MLO2		Java FX MLO2				Basic of User Interface MLO2				9
Week - 5	1		2		2				4		2		11
	Event Driven Programming MLO2		Implement simple event handling to understand event and event sources MLO2		Event Handling MLO2				Event and event sources MLO2		Assessment Task 1: Practical Test MLO1		11
Week - 6	1		2		2				3				8
	Event Driven Programming MLO2		Creating programs to implement event handling including mouse and key events MLO2		Event Handling MLO2				Key event and mouse event MLO2				8
Week - 7	1		2		2				4				9
	User Interface Controls MLO3		Creating Labels, Button, Text Field and Text Area MLO3		GUI MLO3				Labels, Button, Text Field and Text Area MLO3				9
Week - 8								3	4				7
								ILW: User Interface Controls. Creating ComboBox,Listview ScrollBar, CheckBox and RadioButton MLO3	ComboBox,Listview ScrollBar, CheckBox and RadioButton MLO3				7
Week - 9	1		2		2				4				9
	IO Stream MLO3		Creating programs to read and write using Binary IO Stream MLO3		IO files MLO3				Binary IO Stream MLO3				9
Week - 10	1		2		2				4				9
	IO Stream MLO3		Creating programs to read and write using Binary IO Stream MLO3		IO files MLO3				Binary IO Stream MLO3				9
Week - 11	1		2		2				4				9
	Collection MLO4		Create simple programs to implement Set and Map MLO4		Java Collection Framework MLO4				implement Set and Map MLO4				9
Week - 12	1		2		2				3			12	20
	Collection MLO4		Create simple programs to implement Set and Map MLO4		Java Collection Framework MLO4				implement Set and Map MLO4			Assessment Task 3: Program Presentation MLO3	20
Week - 13	1		2	2					3	1		10	19
	Generics		Creating Simple Generic Methods MLO4	Generics MLO4					Creating Simple Generic Methods MLO4	Assessment Task 1: Practical Test MLO1		Assessment Task 3: Program Presentation MLO3	19
Week - 14	1		2		2				3				8
	Generics		Creating Generic Class and objects MLO4		Creating Generic Class and objects MLO4				Creating Generic Class and objects MLO4				8
Week - 15									12				12
									Study Week (preparation of final examination) MLO2				12
Total	13	0	24	2	24	0	0	3	67	3	2	22	

Week - 16									4	2			6
									Final Examination MLO2 MLO3 MLO4 **MLO3 MLO4**	Final Examination **MLO4**			6
Total	13	0	24	2	24	0	0	3	67	3	2	22	

Section 10
Module Description - Reference

Main references supporting the module

No	Author	Year of Publication	Title	Edition	Publisher	ISBN	ISSN	Form Source
1	Y. Daniel Liang	2021-09-28	Introduction to Java Programming and Data Structures, Comprehensive Version, Global Edition	-	Pearson Higher Ed	129240213X; 9781292402130	-	google_books
1	Paul Deitel; Harvey Deitel	2017-02-23	Java How to Program, Early Objects, Student Value Edition	-	Pearson Education	013475185X; 9780134751856	-	google_books

Other Additional Information

No	Author	Year of Publication	Title	Edition	Publisher	ISBN	ISSN	Form Source
1	Robert Sedgewick; Kevin Wayne	2017-04-04	Introduction to Programming in Java	-	Addison-Wesley Professional	0134511603; 9780134511603	-	google_books
1	Walter Savitch	2018-08-07	Java: An Introduction to Problem Solving and Programming, Global Edition	-	Pearson Education	1292247533; 9781292247533	-	google_books

Section 11
Module Description - Approval Details

Effective Study Intake/Semester	**202308**
Revision Number	**1.05**

Special Requirements to deliver the Module

Data Not Available

☐ Please tick ( ) if this module is Latihan Industri/Clinical Placement/Practicum/WBL using 2 weeks, 1 credit for SLT

Approved by SPC	**School of Computer Science**
SPC Approval Date	**Jul 05 2023**
Discipline Code	**-**
Stream	

Information Technology