

## COURSE INFORMATION

<b>School/Faculty:</b>	Computing/ Engineering	<b>Page:</b>	1 of 6
<b>Program name:</b>	Bachelor of Computer Science		
<b>Course code:</b>	SECJ1023/ SCSJ1023	<b>Academic Session/Semester:</b>	20222023/1
<b>Course name:</b>	Programming Technique II	<b>Pre/co requisite (course name and code, if applicable):</b>	Programming Technique I (SECJ1013/ SCSJ1013)
<b>Credit hours:</b>	3		

<b>Course synopsis</b>	This course presents the concept of object orientation and object-oriented programming (OOP) techniques using the C++ programming language. It equips the students with the theory and practice on problem solving techniques using the object-oriented approach. It emphasizes on the implementation of the OOP concepts including encapsulations, associations, inheritance and polymorphism. At the end of this course, students should be able to apply the OOP techniques to solve problems.			
<b>Course coordinator (if applicable)</b>	Dr Jumail Bin Taliba			
<b>Course lecturer(s)</b>	<b>Name</b>	<b>Office</b>	<b>Contact no.</b>	<b>E-mail (@utm.my)</b>
	Ir. Dr. Hazilah Mad Kaidi	F301D		hazilah.kl
	Dr. Mohd Azri Mohd Izhar			mohdazri.kl

### Mapping of the Course Learning Outcomes (CLO) to the Programme Learning Outcomes (PLO), Teaching & Learning (T&L) methods and Assessment methods:

No.	CLO	PLO (CODE)	*Taxonomies and **generic skills	T&L methods	***Assessment methods
CLO1	Analyse problems systematically using object-oriented approaches.	KW Knowledge	*C4	Lecture, active learning, Project-based learning	Project, Mid Term Test, Final Exam
CLO2	Develop programs using object-oriented principles	AP Application	*C5, P2, A2	Lecture, Active-learning, Project-based learning	Exercise, Skill-Based Test, Project, Final Exam
CLO3	Work in a team to develop a medium to complex program as a group mini project, using C++ programming language.	TW Teamworking	*C5, A2, P4 **TW1, TW3	Project-based learning	Project, Peer and Self-Assessment

### Details on Innovative T&L practices:

No.	Type	Implementation
1.	Active learning	Conducted through in-class activities such as pair programming, group discussion
2.	Project-based learning	Conducted through a group project (3 or 4 students per group). The project is divided into several deliverables.

<b>Prepared by:</b> Name: Jumail bin Taliba Signature: Date: 24/08/2017	<b>Certified by:</b> Name: Assoc. Prof. Dr. Radziah Mohamad Signature: Date:
--	---

<b>School/Faculty:</b>	Computing/Engineering	<b>Page:</b>	2 of 6
<b>Program name:</b>	Bachelor of Computer Science		
<b>Course code:</b>	SECJ1023/ SCSJ1023	<b>Academic Session/Semester:</b>	20222023/1
<b>Course name:</b>	Programming Technique II	<b>Pre/co requisite (course name and code, if applicable):</b>	Programming Technique I (SECJ1013/ SCSJ1013)
<b>Credit hours:</b>	3		

**Weekly Schedule:**

<b>Week</b>	<b>Topics</b>	<b>Assessments</b>
<b>Week 1</b> 16-20 Oct 2022	<b>1.0 Introduction</b> 1.1 Overview of Programming Paradigms <ul style="list-style-type: none"> <li>• Procedural Programming</li> <li>• Object-Oriented Programming</li> <li>• Functional Programming</li> </ul> 1.2 Revision on Programming Technique I 1.3 Setting Up Programming Environment	
<b>Week 2</b> 23-27 Oct 2022	1.4 Overview of Object-Oriented Programming Principles <ul style="list-style-type: none"> <li>• Abstractions</li> <li>• Data Hiding</li> <li>• Encapsulations</li> <li>• Associations</li> <li>• Inheritances</li> </ul> 1.5 UML Class Diagram	<i>Project group formation and project Idea</i>
<b>Week 3</b> 30 Oct -3 Nov 2022	<b>2.0 Introduction to Classes and Objects</b> 2.1 Defining Classes 2.2 Creating Objects 2.3 Private Members <ul style="list-style-type: none"> <li>• Why Have Private Members?</li> <li>• Using Private Member Functions</li> </ul> 2.4 Separating Class Specification from Implementation 2.5 Inline Member Functions	<b>Exercise 1</b>
<b>Week 4</b> 6 – 10 Nov 2022	<b>3.0 Constructors and Destructors</b> 3.1 Constructors 3.2 Passing Arguments to Constructors 3.3 Destructors 3.4 Overloading Constructors 3.5 Copy Constructors	<b>Project Deliverable 1: Proposal</b>
<b>Week 5</b> 13 – 17 Nov 2022	<b>4.0 Class and Object Manipulations</b> 4.1 Friend of Classes 4.2 Pointers to Objects 4.3 Arrays of Objects 4.4 Objects and Functions <ul style="list-style-type: none"> <li>• Objects as Function Parameters</li> <li>• Returning Objects from Functions</li> </ul> 4.5 Operator Overloading 4.6 Object Conversions	<b>Exercise 2</b>

<b>School/Faculty:</b>	Computing/Engineering	<b>Page:</b>	3 of 6
<b>Program name:</b>	Bachelor of Computer Science		
<b>Course code:</b>	SECJ1023/ SCSJ1023	<b>Academic Session/Semester:</b>	20222023/1
<b>Course name:</b>	Programming Technique II	<b>Pre/co requisite (course name and code, if applicable):</b>	Programming Technique I (SECJ1013/ SCSJ1013)
<b>Credit hours:</b>	3		

<b>Week 6</b> 20 – 24 Nov 2022	<b>5.0 String Manipulations</b> 5.1 The string Class 5.2 String Comparisons 5.3 String Operators 5.4 String Member Functions	<b>Exercise 3</b>
<b>Week 7</b> 27 Nov –1 Dec 2022	<b>6.0 Associations, Aggregations and Compositions</b> 6.1 Introduction to Associations 6.2 Introduction to Aggregations	<b>Mid-Term Test</b>  <b>Project Deliverable 2: Problem Analysis and Design</b>
<b>Week 8</b> 4 – 8 Dec 2022	<b>MID-SEMESTER BREAK</b>	
<b>Week 9</b> 11 – 15 Dec 2022	6.3 Aggregation Implementations 6.4 Introduction to Compositions 6.5 Composition Implementations	<b>Exercise 4</b>
<b>Week 10 - 11</b> 18 – 22 Dec 2022 25 – 29 Dec 2022	<b>7.0 Inheritance</b> 7.1 Introduction to Inheritance 7.2 Protected Members and Class Access 7.3 Constructors and Destructors in Base and Derived Classes 7.4 Redefining Base Class Functions 7.5 Class Hierarchies 7.6 Multiple Inheritances	<b>Project Deliverable 3: Interim Progress, Week 11</b>
<b>Week 12 - 13</b> 1 – 5 Jan 2023 8 – 12 Jan 2023	<b>8.0 Polymorphisms</b> 8.1 Introduction to Polymorphisms 8.2 Polymorphism and Virtual Member Functions 8.3 Abstract Base Classes and Pure Virtual Functions	<b>Exercise 5</b>
<b>Week 14 - 15</b> 15 – 19 Jan 2023 22 – 26 Jan 2023	<b>9.0 Overview of Advanced Features</b> 9.1 Exceptions 9.2 Templates 9.3 Containers 9.4 Iterators	<b>Exercise 6</b>  <b>Project Deliverable 4: Project Finale, Week 15</b>
<b>Week 16</b> 29 Jan – 2 Feb 2023	<b>REVISION WEEK</b>	
<b>EXAM WEEKS (3 WEEKS: WEEK 17 - 19)</b>		

<b>School/Faculty:</b>	Computing/Engineering	<b>Page:</b>	4 of 6
<b>Program name:</b>	Bachelor of Computer Science		
<b>Course code:</b>	SECJ1023/ SCSJ1023	<b>Academic Session/Semester:</b>	20222023/1
<b>Course name:</b>	Programming Technique II	<b>Pre/co requisite (course name and code, if applicable):</b>	Programming Technique I (SECJ1013/ SCSJ1013)
<b>Credit hours:</b>	3		

**Transferable skills (generic skills learned in course of study which can be useful and utilised in other settings):**

Team working skills.
----------------------

**Student learning time (SLT) details:**

Distribution of student Learning Time (SLT) Course content outline	Teaching and Learning Activities				TOTAL SLT
	Guided Learning (Face to Face)		Guided Learning Non-Face to Face	Independent Learning Non-Face to face	
CLO	L	T	P	O	
CLO 1	15h	7h			23h
CLO 2	13h	7h	12h		31h
CLO 3				3h	2h
<b>Total SLT</b>	<b>28h</b>	<b>14h</b>	<b>12h</b>	<b>3h</b>	<b>54h</b>

L – Lecture; T-Tutorial; P-Practical; O-Others

	Continuous Assessment (Count)	PLO	Percentage	Total SLT
1	Exercises: i. Topic 2: Introduction to Classes and Objects (3%) ii. Topic 4: Class and Object Manipulations (3%) iii. Topic 5: String Manipulations (3%) iv. Topic 6: Associations, Aggregations and Compositions (4%) v. Topic 8: Polymorphism (4%) vi. Topic 10: Overview of Advanced Features (3%)	AP	20	As in CLO2 (14h)
2	Group Project: i. Project Proposal ii. Problem Analysis and Design iii. Interim Progress iv. Project Finale	KW KW AP AP	5 5 5 10	As in CLO1 (3h) As in CLO1 (3h) As in CLO2 (3h) As in CLO2 (8h)
3	Teamworking: i. PSA1 (1%) ii. PSA2 (1%) iii. PSA3 (1%) iv. PSA4 (2%) <i>PSA - Peer and Self-Assessment</i>	TW	5	As in CLO3 (4h)

<b>School/Faculty:</b>	Computing/Engineering	<b>Page:</b>	5 of 6
<b>Program name:</b>	Bachelor of Computer Science		
<b>Course code:</b>	SECJ1023/ SCSJ1023	<b>Academic Session/Semester:</b>	20222023/1
<b>Course name:</b>	Programming Technique II	<b>Pre/co requisite (course name and code, if applicable):</b>	Programming Technique I (SECJ1013/ SCSJ1013)
<b>Credit hours:</b>	3		

Summative and Final Assessments			Percentage	Total SLT
1	Mid Term Test	KW	15	<b>2h</b>
2	Final Exam Pape 1 (Theory)	KW	15	<b>2h</b>
3	Final Exam Paper 2 (Practical)	AP	20	<b>3h</b>
<b>Grand Total SLT</b>				<b>120h</b>

**Special requirement to deliver the course (e.g.: software, nursery, computer lab, simulation room):**

<p>Computer Lab (with internet connections): For Lab Exercises and practical tests  Software: C++ IDE such as Microsoft Visual Studio Code, Dev C++, etc.</p>
---

**Learning resources:**

<p><b>Main references:</b>  Tony Gaddis and Barret Krupnow, (2016), Starting out with C++: From Control Structures through Objects, 8th edition update. Pearson Education.</p> <p><b>Lab Exercise Book:</b>  Faculty of Computing, Programming Technique II – C++ Workbook, 5th edition, 2018. (Compulsory)</p>
---

<p><b>Other references:</b></p> <ol style="list-style-type: none"> <li>1. D. S. Malik,(2015), C++ Programming: From Problem Analysis to Program Design, 7th edition. Cengage Learning.</li> <li>2. Walter Savitch, (2015), Problem Solving with C++. 9th edition. Pearson Education.</li> </ol> <p><b>Online</b>  <a href="http://elearning.utm.my">http://elearning.utm.my</a>  <a href="http://www.cplusplus.com/">http://www.cplusplus.com/</a>  <a href="https://www.tutorialspoint.com/cplusplus/">https://www.tutorialspoint.com/cplusplus/</a></p>
---

**Academic honesty and plagiarism:**

<p>Copying of work (texts, lab results etc.) from other students/groups or from other sources is strictly prohibited. Be warned: students who submit copied work will obtain a mark of <b>zero</b> for the exercises, tests and exams; and disciplinary steps may be taken by the School. It is also unacceptable to do somebody else’s work, to lend your work to them or to make your work available to them to copy.</p>
---

**Other additional information (Course policy, any specific instruction etc.):**

<ol style="list-style-type: none"> <li>1. Attendance is compulsory and will be taken in every lecture session. Student with <b>less than 80%</b> of total attendance is not allowed to sit for final exam.</li> <li>2. Students are required to behave and follow the University’s dressing regulation and etiquette all the time.</li> </ol>
---

<b>School/Faculty:</b>	Computing/Engineering	<b>Page:</b>	6 of 6
<b>Program name:</b>	Bachelor of Computer Science		
<b>Course code:</b>	SECJ1023/ SCSJ1023	<b>Academic Session/Semester:</b>	20222023/1
<b>Course name:</b>	Programming Technique II	<b>Pre/co requisite (course name and code, if applicable):</b>	Programming Technique I (SECJ1013/ SCSJ1013)
<b>Credit hours:</b>	3		

3. Exercises and tutorial will be given in class and some may be taken for assessment. Students who do not do the exercise will lose the coursework marks for the exercise.
4. Exercises must be submitted on the due dates. Some points will be deducted for late submissions. Exercises submitted **three days after** the due date will not be accepted.
5. Make up exam will not be given, except to students who are sick and submit medical certificate confirmed by UTM panel doctors. Make up exam can only be given **within one week** of the initial date of exam.

**Disclaimer:**

No one is allowed to use texts or excerpts from lectures or other teaching and learning activities at Universiti Teknologi Malaysia except for the purpose of his/her studies. In particular, making copies of the texts or excerpts in any form at all for the purpose of publication or distribution is strictly forbidden.

While every effort has been made to ensure the accuracy of the information supplied herein, Universiti Teknologi Malaysia cannot be held responsible for any errors or omissions.