COURSE INFORMATION

Department/ Faculty:	Mechanical Precision Engineering, Malaysia-Japan International Institute of Technology	**Page:**	1 of 4	
		Student learning Times (Hours) :		120
		Revision :		D
Course code:	SMJP 1043	**Academic Session/Semester:**		20182019 / 2
Course name:	Programming for Engineers	**Pre/co requisite (course name and code, if applicable):**		
Credit hours:	3			

Course synopsis	This course is designed to expose students to the development of programming skill using a C++ language, which is suitable for the current computer operating system. It emphasizes on the general concept of computer programming that includes steps of problem solving using computer, algorithm and programming logic tools. Students will be introduced to variables and operators, mathematic operations, commands and functions, program control structures, modularization, and input and output files. Examples, assignments, hands-on exercises and group projects related to various engineering fields will be given to the students. At the end of the course, students should be able to plan, analyse and write computer programs for basic engineering applications. This course embraces authenticity of generic skills (thinking skills) when engaging in the process of completing the tasks.			
Course coordinator (if applicable)	Dr Zainudin Bin A. Rasid			
Course lecturer(s)	**Name**	**Office**	**Contact no.**	**E-mail**
	Dr Zainudin Bin A. Rasid	6-45-01	0132773550	arzainudin.kl@utm.my
	AP. Dr Sheikh Ahmad Zaki bin Shaikh Salim	10.38.01	03-22031483	sheikh.kl@utm.my
	Dr Nelidya Bte Md. Yusoff	14-8-1	0126998806	nelidya.kl@utm.my

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

No.	CLO	PLO EAC / UTM	Weight (%)	Taxo. & generic skills*	T&L methods	WP	WK	EA	Assessment methods
CO1	Define basic computer programming development procedures.	1 / KW	10	C3	Lecture & Practical	WP 1	1, 4		Q
CO2	Apply logic design tools to produce a computer program.	3 / THDS	40	C3, C4, C5	Lecture & Practical	WP 1	5		ASG, T

Prepared by: Name: Dr Zainudin Bin A. Rasid Signature: Date: 20-08-2018	Certified by: Name: Signature: Date:

No.	CLO	PLO EAC / UTM	Weight (%)	Taxo. & generic skills*	T&L methods	WP	WK	EA	Assessment methods
CO3	Analyse and develop computer program using programming language.	2 / THPA	30	C5	Lecture & Practical		1, 4		LT
CO4	Develop computer program to solve relevant basic engineering problems.	2, 10 / THPA, TW	20	C5, TW1	Problem based learning		1, 8		PR

Refer *Taxonomies of Learning and **UTM's Graduate Attributes, where applicable for measurement of outcomes achievement

***T – Test; Q – Quiz; ASG –Assignment; PR – Project; Pr – Presentation; F – Final Exam; R-Report; PR-Peer Review etc.

Details on Innovative T&L practices:

No.	Type	Implementation
1.	Lecture	2 hours / week X 14 weeks = 28 hours
2.	Practical (Hands-on)	3 hours / week X 14 weeks = 42 hours
3.	Non-face-to-face learning	12 hours
4.	Revision	14 hours
5.	Assessment Preparations	12 hours
6.	Continuous Assessment	12 hours

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

Skill in analysing information, Thinking Critically, solving Problems, disseminating application to engineering

Weekly Schedule:

Week 1	Introduction to Computer and Programming, Introduction to C++
Week 2	Expressions and Interactivity
Week 3	Making Decisions **Assignment 1**
Week 4	Making Decisions Quiz 1 Assignment 2
Week 5	Looping & Files
Week 6	Looping & Files Assignment 3
Week 7	Test 1 (Lab and Writing)
Week 8	Mid-Semester Break
Week 9	Function Assignment 4
Week 10	Function
Week 11	Arrays Quiz 2 Project proposal submission
Week 12	Arrays Assignment 5
Week 13	Test 2 (Lab and writing)
Week 14	Project Preparation

Week 15	Project Presentation

Student learning time (SLT) details:

Distribution of student Learning Time (SLT) Course content outline	Guided Learning (Face to Face) Lecture Practical/Tutorial/Studio Student-Centered learning				Teaching and Learning Activities: Guided Learning Non-Face to Face Revision	Teaching and Learning Activities: Independent Learning Non-Face to face Self-Directed learning Assessment Preparation Revision	TOTAL SLT
CLO	**L**	**T**	**P**	**O**			
CO1	6					5	11
CO2	10			5	10	15	40
CO3	10			2	11	10	33
CO4	5			4		12	21
Total SLT	**31**			**11**	**21**	**41**	**105**

ASSESSMENT DETAILS

Continuous Assessment		CLO	PLO EAC/UTM				Taxo	Total SLT
Components	Percentage		1/KW	2/THPA	3/THDS	10/TW	Gen.	
Quiz 1 & 2	10%	1	10				C3	1
Assignment	10%	2			10		C5	5
Test 1 & 2 (Writing)	30%	2			30		C4,C5	3
Test 1 & 2 (Lab)	30%	3		30			TH3	3
Final Assessment								
Final Project	20 %	4		10		10	C4,C5, TH3	3
Total Marks	100 %							
Total SLT Continuous Assessment								**15**
Grand Total SLT								**120**

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

Dev C++ Software

Learning resources:

Text book (if applicable)

1. Tony *Gaddis*, Starting Out with *C++*: From Control Structures through Objects, 7/e, Pearson,2011, ISBN.
2. K. N. King, C Programming: A Modern Approach, 2nd edition , W. W. Norton, 2008.
3. Stephen Kochan, Programming in C, 3rd Edition, Sams, 2004.
4. Deitel and Deitel: C++ How to Program, 3rd Edition, Prentice Hall 2002.
5. Capper, D., Introducing C++ for Scientists, Engineers and Mathematicians, Springer, 2001, 1852334886, Y.
6. Delores M. Etter, Engineering Problem Solving with C++, 2nd Edition, US Naval Academy, ISBN-10 0136011756.

Academic honesty and plagiarism:

Cheating is not only dishonest, but also self-destructive. Some of the principles of academic honesty that are especially important in this courses are:

- Plagiarism is a very serious violation. All the writing in your documentation and/or reports must be your own work. You may not copy sentences or paragraphs from books, web pages, other students, or any other source. If you quote or use anything written by anyone else, you must indicate very clearly that it is a quotation ***and*** you must provide a full citation.
- All the programming code that you claim credit for (implicitly or explicitly) must be your own creation. If you use software written by anyone else, you must disclose this very clearly both in your code and in all accompanying documentation and reports.
- Tables and figures of programming results that show how your programs run, must be genuine and not misleading. It may happen that some of your code or algorithms do not work correctly. In this case you must mention and explain this situation in documentation and reports.
- If you work in a team on any assignment or project, and there is a case of academic dishonesty, then all members of the team will be assumed to be equally responsible and will be subject to the same penalties. If you work in a team, it is your responsibility to make sure that your partners are as honest as you are, and that they are well-informed about what is permissible.

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

Disclaimer: