COURSE OUTLINE

Course Code: SCSJ1023
Course Name: Programming Technique II
Total Contact Hours: 56 hours
Course Pre-requisite: Prog. Technique I (SCSJ1013)

SYNOPSIS

This course equips the students with theory and practice on problem solving techniques by using two approaches, namely the structured approach and the object oriented approach. The first part of the course provides students with basic skills to program in Linux platform and advanced concepts in structured programming techniques including pointers, advanced file operations and strings and its manipulation. The second part of the course is to provide students with object-oriented techniques such as class, objects, overloading, aggregation, composition, inheritance, polymorphism, virtual functions, exception handling and templates.

LEARNING OUTCOMES

By the end of the course, students should be able to:

No.	Course Learning Outcome	Programme Learning Outcome(s) Addressed	Assessment Methods
1.	Analyze problems systematically using structured and object oriented approaches.	PO2 (C4, P3, A2)	LE, Q, A, T, F
2.	Construct or develop C++ programs correctly using advanced structured and object oriented features such as pointers, files, aggregation and inheritance.	PO1 (C3, P3, A2)	LE, Q, A, T, PR, F
3.	Solve problems in a given time frame using C++ programming language and tools.	PO1 (C3, P3, A2)	Sbt, A, T
4.	Work in a team to develop a medium to complex program as a group mini project, using C++ programming language.	PO6 (TS1- TS3)	PR, A, Pr, Peer
5.	Communicate mini project deliverables in writing and oral presentation. (T – Test ; Q – Quiz; LE – Lab exercise ; Sbt - Skill-Based Test; A – Assignment; Peer – Peer assessment; PR – Project ; Pr – Presentation, F – Final Exam)	PO6(CS1, CS3, CS4)	Pr, A, PR

STUDENT LEARNING TIME

Teaching and Learning Activities			Student Learning Time (hours)	
Face to face Learning	• Lecturer Centered	Lecture	28	
	• Student Centered	- Practical/Lab/Tutorial	24	
		- Student Centered Activity	4	
	• Others		0	
	Sub Total			56
Self Learning	• Non Face to face or Student Centered Learning (SCL)		16	
	• Revision		14	

	• Assessment Preparation	19	
	• Others	0	
	Sub Total		49
Formal Assessment	• Continuous Assessment	12	
	• Final Examination	3	
	• Others	0	
	Sub Total		15
TOTAL SLT			**120**

TEACHING METHODOLOGY

Lecture and Discussion, Lab Activities, Co-operative Learning, Mini Project, Presentation, Independent Study

WEEKLY SCHEDULE

Week	Topics	Activities/hours
Week 1	**1.0 Introduction to LINUX and Editors** 1.1 LINUX introduction 1.1.1 Virtual Machine (vmWare) 1.1.2 X windows 1.1.3 Basic commands 1.1.4 C++ Programming in LINUX environment (GNU Compiler Collection) 1.2 Editors 1.2.1 vi Editor 1.2.2 emacs Editor	Lecture: 2, Lab: 2 *Lab Exercise 0: Revision topics* - *Functions* - *Arrays* - *Struct*
Weeks 2-3	**2.0 Pointers** 2.1 Address of a Variable 2.2 Pointer Variable 2.3 The Relationship Between Arrays and Pointers 2.4 Pointer Arithmetic 2.5 Initializing Pointers 2.6 Comparing Pointers 2.7 Pointers as Function Parameters 2.8 Dynamic Memory Allocation 2.9 Returning Pointers from Functions 2.10 Pointers to Structures	Lecture: 4, Lab: 4 *Lab Exercise 1: (wk. 3 – Pointers)*
Week 4	**3.0 String and String Manipulation** 3.1 Character Testing 3.2 Character Case Conversion 3.3 The C-Strings 3.4 String/Numeric Conversion Functions 3.5 The C++ string Class	Lecture: 2, Lab: 2 *Skill-based Test 1 (Pointers)* *Assignment 1 (Ptrs and/or Strings)*
Week 5	**4.0 Advanced File Operations** 4.1 File Operations 4.2 File Output Formatting 4.3 Passing File Stream Objects to Functions 4.4 Member Functions for Reading and Writing Files 4.5 Multiple Files 4.6 Binary Files 4.7 Creating Records with Structures 4.8 Random-Access Files 4.9 Opening a File for Both Input and Output	Lecture: 2, Lab 2 *Lab Ex. 2 (Ptrs, Strings, Adv. Files)*

Week 6	**5.0 Introduction to Class** 5.1 Procedural and Object-Oriented Programming 5.2 Introduction to Classes 5.3 Defining an Instance of a Class 5.4 Private Members 5.5 Separating Class Specification from Implementation 5.6 Inline Members Functions 5.7 Constructors 5.8 Passing Arguments to Constructors	Lecture: 2, Lab 2 *Test 1 (Pointers, Strings, Adv. File Operations)* *Project briefing*
Week 7	5.9 Destructors 5.10 Overloading Constructors 5.11 Private Members Functions 5.12 Arrays of Objects 5.13 The Unified Modeling Language (UML)	Lecture: 2, Lab 2 *Lab Ex. 3 (UML Diagrams)* **Phase 1 : Proposal due*
Weeks 8-9	**6.0 Class and Object Manipulation** 6.1 Instance and Static Members 6.2 Friends of Classes 6.3 Memberwise Assignment 6.4 Copy Constructors 6.5 Operator Overloading 6.6 Object Conversation	Lecture: 4, Lab 4 *Skill-based Test 2 (Intro. To Class)* *Lab Ex. 4 (Obj. & Class Manipulation)* **Phase 2 : Design due (wk. 9)*
Week 10	**7.0 Aggregation and Composition** 7.1 Introduction to Aggregation 7.2 Aggregation Implementation 7.3 Introduction to Composition 7.4 Composition Implementation	Lecture: 2, Lab 2 *Test 2 (Obj. & Class Manipulation)* *Assignment 2 (Class Relationships)*
Week 11	**8.0 Inheritance** 8.1 Introduction to Inheritance 8.2 Protected Members and Class Access 8.3 Constructors and Destructors in Base and Derived Classes 8.4 Redefining Base Class Functions 8.5 Class Hierarchies	Lecture: 2, Lab 2 *Lab Ex. 5 (Inheritance)* **Phase 3 : Initial results due*
Week 12	**9.0 Polymorphism and Virtual Function** 9.1 Introduction to Polymorphism 9.2 Polymorphism Implementation 9.3 Introduction to Virtual Function 9.4 Virtual Function Implementation 9.5 Abstract Base Class and Pure Virtual Function	Lecture: 2, Lab 2 *Skill-based Test 3 (Aggr., Comp, & Inheritance)*
Week 13	**10.0 Exception Handling and Templates** 10.1 Introduction to Exception Handling 10.2 Exception Handling Implementation 10.3 Introduction to Templates 10.4 Templates Implementation	Lecture: 2, Lab 2 *Lab Ex. 6 (Polymorphism)*
Week 14	**Mini Project Presentation**	**Phase 4 : Final results due*
	STUDY WEEK (7 June – 9 June 2015)	
	EXAMINATION WEEKS (10 June – 27 June 2015)	

REFERENCES :

Courses Notes:

Tony Gaddis and Barret Krupnow, (2012), *Starting out with C++: From Control Structures through Objects*, 7th edition update. Pearson Education.

Lab Book:

Faculty of Computing, *Programming Technique II – C++ Workbook (English – Malay)*, 4th edition, 2015.

Other References:

1. D. S. Malik,(2012), *C++ Programming: From Problem Analysis toProgram Design*, 6th edition. Cengage Learning.
2. Walter Savitch, (2012), *Problem Solving with C++*. 8th edition. Addison-Wesley.
3. H.M. Deitel and P.J. Deitel, (2012), *C++ How to Program (Late Objects)*. 8th edition. Pearson Education.

GRADING

No.	Assessment	Number	% each	% total
1	Assignments	2 (individual)	5%	10
2	Lab Exercises	minimum 4	2.5%	10
3	Skill-based Test	3	3%, 3%, 4%	10
4	*Group Project (Phase 1-4)	1	10%	10
5	Project Presentation (individual)	1	5%	5
6	Test 1 (written - theory)	1	15%	15
7	Test 2 (lab - programming)	1	10%	10
8	Final Exam (theory, no lab)	1	30%	30
	Overall Total			**100**