

COMPUTER GAME PROGRAMMING 1

BITE 1513

SEMESTER 1

SESI 2016/2017

BITE 1513 PENGATURCARAAN PERMAINAN KOMPUTER 1 [3]

TYPE OF COURSE: C

EDITION : 1

UPDATED DATE : 26-8-2016

1.0 LEARNING OUTCOMES

Upon completing this subject, students should be able to:

1. Explain the key components of the programming language C++. (C3)
2. Demonstrates the core principles of C++ programming. (P2, CTPS3)
3. Develop a simple game program using C++ (P3, A3, LL1)

2.0 SYNOPSIS

The course is conducted to introduce the basic concepts of C + + programming includes variable declaration, loops and conditional repetition, arrays, pointers, functions and basic object-oriented programming. Basic object-oriented programming involves classification of objects, polymorphism and inheritance. The course also exposes the concept of input-output and file manipulation some advanced topics such as C + + virtual functions, static overloading and operation. The purpose of this course is to provide basic knowledge of C + + so that students can display core competencies C + + and is able to produce a simple game using C + +.

3.0 PRE-REQUISITE

-None-

4.0 PRACTICAL

The use of Microsoft Visual Studio [Visual C++] for simple game development.

5.0 REFERENCES

- [1] Dawson, M. (2014) Beginning C++ Through Game Programming. Cengage Learning PTR; 4 edition (June 23, 2014)
- [2] Malik, D. S. (2014). C++ Programming: From Problem Analysis to Program Design. Cengage Learning; 7 edition (February 7, 2014)
- [3] Horstmann, C. S. and Budd, T. A. Big (2008). C++, 2nd Edition. John Wiley.
- [4] Conger, D. (2007). Creating Games in C++: A Step-by-Step Guide. New Riders Games.
- [5] Thorn, A. (2007). Introduction to Game Programming with C++. Jones & Bartlett Publishers.

6.0 COURSE IMPLEMENTATION

- i. Lecture
 - 2 hours per week for 14 weeks (Total = 28 hours)
- ii. Lab
 - 2 hours per week for 14 week (Total = 28 hours)

7.0 COURSE EVALUATION

Assessment Method	LO1	LO2	LO3	Scheme, Rubric/guideline
Quiz (2) = 5%	Q1(3%)	Q2(2%)		
Project 1 = 10%		P1(5%)	P1(5%)	
Lab Assessment = 10%		(5%)	(5%)	
Project 2 = 10%			P2(10%)	
Presentation (1)= 5%	5%			
Lab Test (2) = 20%		LT1(10%)	LT2(10%)	
Mid Term (1) = 10%	(5%)	(5%)		
Final (1) = 30%	(18%)	(12%)		
Total	31%	39%	30%	

8.0 STUDENT LEARNING TIME (SLT)

LEARNING ACTIVITIES	STUDENT LEARNING TIME (BITE 1513)											
	GUIDED LEARNING TIME						INDEPENDENT LEARNING TIME					
	OFFICIAL CONTACT HOURS	FREQ	TOTAL	GUIDED LEARNING HOURS	FREQ	TOTAL	SELF STUDY HOURS	FREQ	TOTAL	ASSESSMENT TIME	FREQ	TOTAL
Lecture	2	14	28	-	-	-	2	14	28	-	-	-
Laboratory + Report	2	14	28	-	-	-	1	14	14	-	-	-
Tutorial	0	0	0	-	-	-	0	0	0	-	-	-
Quiz	-	-	-	-	-	-	0.5	2	1	0.25	2	0.5
Theoretical Test (Midterm)	-	-	-	-	-	-	2	1	2	1	1	1
Lab Test	-	-	-	-	-	-	2	1	2	1	1	1
Discussion	-	-	-	-	-	-	-	-	-	-	-	-
Mini Project - Group	-	-	-	-	-	-	4	1	4	-	-	-
Mini Project - Individual	-	-	-	-	-	-	4	0	0	-	-	-
Assignment - Group	-	-	-	-	-	-	2	1	2	-	-	-
Assignment - Individual	-	-	-	-	-	-	2	1	2	-	-	-
Presentation - Group	-	-	-	-	-	-	1	1	1	0.5	1	0.5
Presentation - Individual	-	-	-	-	-	-	1	1	1	0.5	1	0.5
Final	-	-	-	-	-	-	8	1	8	2	1	2
TOTAL	56			0			65			5.5		
GRAND TOTAL	126.5											
TOTAL CREDIT	3.1625											

9.0 DETAILED SYLLABUS AND TEACHING PLAN

Week	Session	Contents	References	Delivery Method
1	Lecture 1	<p>INTRODUCTION TO GAMES PROGRAMMING?</p> <p>Lecture Content</p> <ol style="list-style-type: none"> 1. What is programming language? 2. The difference between a programming language and language 3. Tools that is used for programming and authoring 		<p>Explain course outline and delivery method</p> <p>Lecture</p>
2	Lecture 2	<p>PROBLEM SOLVING</p> <p>Lecture Content</p> <ol style="list-style-type: none"> 1. Software development methodology and software life cycle 2. Requirement specification, Problem Analysis, System Design and System Development <p>Laboratory</p> <p>To explore and differentiate which one the tools that can be categorize in programming language and authoring language.</p> <p>Developing algorithm using pseudo code and flow charts</p>		<p>Lecture</p> <p>Group Project 1 (Project Proposal)</p>
3	Lecture 3	<p>INTRODUCTION TO C++ PROGRAMMING</p> <p>Lecture Content</p> <ol style="list-style-type: none"> 1. Type, Variables and Standard I/O 2. Working with std namespace 3. Using Arithmetic Operators 4. Declaring the variables 5. Performing Arithmetic Operations with variables 6. Working with constants <p>Laboratory</p> <p>Compile and execute basic programming using C++ and produce games “Lost Fortune”</p>	Chapter 1	Lecture

4	Lecture 4	TRUTH, BRANCING AND THE GAME LOOP Lecture Content <ol style="list-style-type: none"> 1. Understanding Truth 2. Using the if statement 3. Using the else statement 4. Using the switch statement 5. Using while loop statement 6. Using do Loops 7. Understanding the Game Loop Laboratory To produce games “My Guess Number”	Chapter 2	Lecture
5	Lecture 5	FOR LOOPS, STRING AND ARRAYS Lecture Content <ol style="list-style-type: none"> 1. Using the Loops 2. Understanding Objects 3. Using String Objects 4. Using Arrays 5. Understanding C-Styles Strings 6. Using Multidimensional Arrays Laboratory <ul style="list-style-type: none"> - MIDTERM EXAM (Chapter 1 – 5) 	Chapter 3	Lecture
6	Lecture 6	THE STANDARD TEMPLATE LIBRARY Lecture Content <ol style="list-style-type: none"> 1. Introducing the Standard Template Library 2. Using Vector 3. Using Iterators 4. Using Algorithms 5. Using Vector Performance Laboratory To produce games “Hangman”	Chapter 4	Lecture

7	Lecture 8	FUNCTIONS Lecture Content <ol style="list-style-type: none"> 1. Creating Functions 2. Using Parameters and Returns Values 3. Understanding Software Reuse 4. Working With Scopes 5. Using Global Variables and Constants 6. Overloading Functions 7. In lining Functions Laboratory To produce games “Mad Lib”	Chapter 5	Lecture Group Project 2 (A Game)
8		MIDTERM BREAK		
9	Lecture 9	REFERENCES Lecture Content <ol style="list-style-type: none"> 1. Using References 2. Passing References to Alter Arguments 3. Passing References for Efficiency 4. Returning References Laboratory Lab Test 1	Chapter 6	Lecture
10	Lecture 10	POINTERS Lecture Content <ol style="list-style-type: none"> 1. Understanding Pointer Basics 2. Understanding Pointers and Constants 3. Passing and Returning Pointers 4. Relationship Between Pointers and Arrays Laboratory To produce games “Tic-Tac-Toe”	Chapter 7	Lecture

11	Lecture 11	CLASSES Lecture Content <ol style="list-style-type: none"> 1. Defining New Types 2. Using Constructors 3. Setting Member Access Levels 4. Using Static Data Members and Member Functions Laboratory To produce games "Critter Caretaker"	Chapter 8	Lecture
12	Lecture 12	ADVANCED CLASSES AND DYNAMIC MEMORY Lecture Content <ol style="list-style-type: none"> 1. Using Aggregation 2. Using Friend Functions and Operator Overloading 3. Dynamically Allocating Memory 4. Working with Data Members and the Heap Laboratory Lab Test 2	Chapter 9	Lecture
13	Lecture 13	INHERITANCE AND POLYMORPHISM Lecture Content <ol style="list-style-type: none"> 1. Introducing Inheritance 2. Controlling Access under Inheritance 3. Control and Overriding Base Class Member Functions 4. Introducing Polymorphism Laboratory Presentation of Project 1	Chapter 10	Lecture
14	Lecture 14	OPENGL WITH C++ PROGRAMMING SIMPLE Laboratory Simple Programming with OpenGL		Lecture
15		REVISION		Lecture