Module (Advanced Programming) Catalogue

2023-2024

College/ Institute	Erbil Technology College	
Department	Information & Communication Technology Engineering (ICTE)	
Module Name	Advanced Programming	
Module Code	ADP603	
Degree	Technical Diploma [x] Bachelor [x] High Diploma [ ] Master [ ] PhD [ ]	
Semester	Sixth Semester	
Qualification	Ph.D.	
Scientific Title	Lecturer	
ECTS (Credits)	6	
Module type	Prerequisite [ ] Core [x] Assist. [ ]	
Weekly hours	4	
Weekly hours (Theory)	(2)hr Class	(91)Total hrs Workload
Weekly hours (Practical)	(2)hr Class	(71)Total hrs Workload
Number of Weeks	15	
Lecturer (Theory)	Chiman haidar salh	
E-Mail & Mobile NO.	Chiman.salh@epu.edu.iq	
Lecturer (Practical)	Chiman haidar salh	
E-Mail & Mobile NO.	Chiman.salh@epu.edu.iq	
Websites		

Course Book

Course Description	This course could be considered as the main course where basic concepts will be explained. These concepts are frequently repeated in other programs. Electronic devices are designed and operated by one of the operating systems which developing by one of the programming languages such as C, C++, C#, Visual Basic, and so on. Therefore any operating system or any application program that you want to create for the purpose of executing any specific job must be made by one of those programming languages classified as a high-level language. C++ is an important programming language to create an application program, Therefore you need to know or learn the basic concepts in the C++ language to create this kind of program. Then you have to have a good background in C++ and Object Oriented Programming because this will be your fundamental step and help you to learn easily. Finally, programming is considered the backbone of computer science.
Course objectives	• Write easier-to-read and easier-to-code using operator overloading • Perform file input-output and describe the concepts of persistent objects • Understand issues in using virtual base classes and multiple inheritances • Better utilize the features of inheritance and polymorphism in program design • Understand how templates can reduce code replication when used properly • To be comfortable using and learning different programming languages (C, C++, Java, Perl) and choosing the appropriate one for a given task and use tools and write programs to assist in developing programs. • Describe when/how to use run-time type identification, mutable keywords/other language features
Student's obligation	The Students should be attendant in class at less than 1:30 hours during lecturing and to pass this course should be fulfilled the following requirements: 1. The student has to submit almost all assignments, essays and reports and also. 2. The student must be passing the exams and quizzes which have been done during study year. 3. Students attending classes regularly. 4. Group work. 5.Doing assignments.

	6.Class activities.
Required Learning Materials	The ways that we are using in our teaching for this course are: 1. Data show. 2. White Board. 3. Word Documents. 4. Notebook. 5. Group activity 6. Computer Lab

Evaluation	**Task**		**Weight (Marks)**	**Due Week**	**Relevant Learning Outcome**
	Paper Review		0		
	Assignments	Homework	5		
		Class Activity	2		
		Report	5		
		Seminar	5		
		Essay	0		
		Project	0		
	Quiz		8		
	Lab.		10		
	Midterm Exam		25		
	Final Exam		40		
	Total		100		

Specific learning outcome:	• This course will teach the student how to solve problems arising from subtleties of the C++ language as well as techniques for improving performance and efficiency. Students are invited to bring their current ideas and question to the classroom for discussion. • On successful completion of the module, students should be able to demonstrate a hands-on understanding of advanced programming topics
Course References:	**1. Internet recourse**

Course topics (Theory)	**Week**	**Learning Outcome**
Introduction & Refresher of OOP	1	review of C++ and classes
C++ Pointers, Array Name as Pointers	2	Pointers

C++ Polymorphism	4	Operator Overloading, Unary Operators, function Over loading
C++ Virtual Functions	5	Pure and Virtual Function
Exception Handling	6	Handling (try, catch, throw)
Catching Class Types, File Handling through C++ Classe	7	I/O File Handling
C++ Circular	8	template < typename T >
Dynamic Containers (STL Components)	9	Standard Template Library (STL)
C++ Lambda, Creating a Lambda Expression in C++	10	Creating a Lambda Expression in C++
Multitasking in C++, C++ Header Files (Include file)	11	C++ Header Files (Include file)
C++ Header Files (Include file), Multithreading in C++	12	C++ Header Files
Smart Pointers in C++	13	Smart Pointers in C++
Type of 'this' Pointer in C++	14	Type of 'this' Pointer in C++
Practical Topics	**Week**	**Learning Outcome**
Introduction & Refresher of OOP	1,2	review of C++ and classes
C++ Pointers, Array Name as Pointers	3	Pointers
C++ Polymorphism	4	Operator Overloading, Unary Operators, function Over loading
C++ Virtual Functions	5	Pure and Virtual Function
Exception Handling	6	Handling (try, catch, throw)
Catching Class Types, File Handling through C++ Classe	7	I/O File Handling
C++ Circular	8	template < typename T >
Dynamic Containers (STL Components)	9	Standard Template Library (STL)
C++ Lambda, Creating a Lambda Expression in C++	10	Creating a Lambda Expression in C++
Multitasking in C++, C++ Header Files (Include file)	11	C++ Header Files (Include file)
C++ Header Files (Include file), Multithreading in C++	12	C++ Header Files

Smart Pointers in C++	13	Smart Pointers in C++
Type of 'this' Pointer in C++	14	Type of 'this' Pointer in C++

Questions Example Design

Q1/ Write down the file I/O step process and explain by examples.

File I/O Step Process

1. Include the header file (fstream/ istream/ ostream)

#include<fstream>

2. Declear the file stream variables:

ifstreammiFile; // Input file

ofstream iFile; // Output file

3. Associated the file stream variable with the input/ output sources.

iFile .open ("file. Txt"); // open the input file

oFile. Open(" file .txt); // open the output file

4. Cheak the files opened:

if (! iFile.is_open()) {……}

if (! iFile.is_open()) {……}

5. Use the file stream variable with >> ,<< or other input/ output function:

iFile>> variable/literal / manipulator;

iFile<< variable / literal / manipulator;

6. Close the file:

iFile.close( ); // Open the input file

oFile.close( ); // Close the output file

Q2/ **Choose the most appropriate statement for the following question.**

1. What are the different types of exceptions?

a) 1 b) 2 c) 3 d) 4

2. Which is used to handle the exceptions in C++?

a) catch handler b) handler c) exception handler d) throw

3. Which type of program is recommended to include in try block?

a) static memory allocation b) dynamic memory allocation

c) const reference d) pointer

4. Which header file is required to use file I/O operations?

a) <ifstream> b) <ostream> c) <fstream> d) <iostream>

5. Used to tell the compiler that the file will be opened for reading from.

a) ios::in b) ios::out c) ios::app d) ios::trunc

1.b
2.c
3.b
4.c
5.a

Q3/ What is an exception-handling mechanism? Also, write down the C++ program**.**

- C++ exception handling mechanism is basically built upon three keywords namely, try, throw and catch.
- Try block hold a block of statements which may generate an exception.
- When an exception is detected, it is thrown using a throw statement in the try block.
- A try block can be followed by any number of catch blocks.

```
#include<iostream>
using namespace std;
void main( )
{
        int x, y,z;
        cout << "enter two number" << endl;
        cin >> x >> y;
        try
        {
                if (y != 0)
```

```
        {
                z = x / y;
                cout << endl << z;
        }
        else
        {
                throw(y);
        }
}
catch (int y)
{
        cout << "exception occurred : y = " << y << endl;
}
system("pause"); }
```

Extra notes:

Please take this note into consideration:
Making the topics covered in the semester compatible with reality of the educational process due to the national and religious holiday or other reasons. For instance, covering more than 80% of the prescribed subjects to improve the scientific level of students and preserve the standardization of diploma programs.

External Evaluator

I have been reviewed this course book, and it's perfect and fit for this subject at the level of institute student, so I have no suggestion.

Zanyar Shwan Ahmed
Lecture (Erbil Technology College) Information Communication Technology Engineering department