

Course Book

Course Description	<p>This course teaches the student to apply Advanced procedural programming concepts to the programming language This course introduces the student to object-oriented programming through a study of the concepts of program specification and design, and coding and testing using a modern software development environment. Students learn how to write programs in an object-oriented high level programming language. Topics covered include fundamentals of problem solving, programming concepts, classes and methods, control structures, arrays, and strings. Throughout the semester, problem solving skills will be stressed and applied to solving computing problems. Weekly laboratory experiments will provide hands-on experience in topics covered in this course.</p>
Course objectives	<p>The objective of this class is to expose the student to procedural programming using C++ and to increase the depth of students' knowledge about several implementation issues. Knowing C++ will be useful in the students' jobs in IT organizations as developers or managers because it will enable them to code efficiently, communicate effectively with colleagues and understand and improve software development practices in their organizations</p>
Student's obligation	<ol style="list-style-type: none">1- Students should attend to the class with Laptop.2- Students should do (achieve) activities like Homework, quizzes, Assignments, Projects and Lab. Reports.3- Students should participate and interact with the topics.
Required Learning Materials	<p>Code::Blocks Version 20.03 Software Data Show Projector, Laptop and hard copy of lecture</p>

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

Specific learning outcome:

- At the end of the course, a student completing this course will be able to:
- 1) Demonstrate the use of numeric arrays, references, pointers and memory address,.
 - 2) Design and implement code that includes the reuse of both existing code and calling functions in the C++ libraries.
 - 3) To learn how to design C++ classes for code reuse.
 - 4) To learn how to implement copy constructors and class member functions.
 - 5) To understand the concept of data abstraction and encapsulation.
 - 6) To learn how to overload functions and operators in C++.
 - 7) To learn how containment and inheritance promote code reuse in C++.
 - 8) To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
 - 9) To learn how to design and implement generic classes with C++ templates.
 - 10) To learn how to use exception handling in C++ programs.

<p>Course References:</p>	<p>1- Prinz&Prinz, Ulla Kirch &Peter, A Complete Guide to Programming C++, Jones and Barlett Publishers, 2002.</p> <p>2- Lee, Mark, C++ Programming for the Absolute Beginner, Second Edition, 2009</p> <p>3- Salman, Nassir, C++ Programming With 469 Solved Problems, Second Edition, 2014.</p> <p>4- مطيع وغيرهم, البرمجة بلغة C++ , الطبعة الاولى , 2006.</p>	
<p>Course topics (Theory) & Practical Topics</p>	<p>Week</p>	<p>Learning Outcome</p>
<p>Review of the Previous Semester</p>	<p>1</p>	<ul style="list-style-type: none"> ✓ Students will introduce to the programming generally and C++ specially. ✓ An introduction about the code blocks software and apply C++ codes on it. ✓ Describe the functions of an IDE. ✓ Use an IDE to compile, load, save, and debug a C/C++ program ✓ Students will learn how to output (print) and make new line by using C++ codes (cout<<, endl or \n). ✓ Students will learn commenting in C++ and practice various types of comments ✓ (// , /* */). ✓ Learning a general idea about variables in C++ with syntax of it and how to use it correctly with keywords.
<p>Review of the Previous Semester</p>	<p>2</p>	<ul style="list-style-type: none"> ✓ Students will learn how to display variables in C++. ✓ Students will be able to use identifiers & constants in C++. ✓ Students will be able to recognize among data types. ✓ Students will learn about (integer,float,double,char and string) types. ✓ Students will be able to swap numbers among variables.

C++ Operators 1	3	✓ Students will gain a general idea about arithmetic and assignment operators.
C++ Operators 2	4	✓ Students will comprehend the comparison and logic operators.
C++ Strings	5	✓ Students will understand about the types of strings and will introduce to String Concatenation, Numbers and Strings, String Length, Access Strings, Change String Characters.
C++ Strings, Maths & Booleans	6	✓ Students will know about User Input Strings and String Namespace. ✓ Students will understand about the C++ Math and C++ Boolean, Boolean Values, Boolean Expressions.
C++ Conditions and If Statements	7	✓ Students will be able to use if conditions, if..else, if...else if....else Statements.
C++ Switch , Break & Default	8	✓ Students will be able to use switch statements and switch with break, default keywords.
While & Do While Loops	9	✓ Students will be able to use while and do while loops with conditions.
For Loop , Break & Continue	10	✓ Students will be able to use for loops. ✓ Students will comprehend the mix between for loop or while loop & break or continue.
Arrays	11	✓
References, Memory Address, Pointers, Dereference, Modify Pointers	12	

Questions Example Design

Q1) Write a program in C++ to display n terms of natural number and their sum.

Ans1)

```
#include <iostream>
```

```

using namespace std;
int main()
{
int n,i,sum=0;
cout << "\n\n Display n terms of natural number and their sum:\n";
cout << "-----\n";
cout << " Input a number of terms: ";
cin>> n;
cout << " The natural numbers upto "<<n<<"th terms are: \n";
for (i = 1; i <= n; i++)
{
cout << i << " ";
sum=sum+i;
}
cout << "\n The sum of the natural numbers is: "<<sum << endl;
}

```

Q2) Write C++ program to find the maximum of three numbers.

Ans2)

```

#include<iostream.h>
int main()
{
int x,y,z;
cout<<"input three numbers\t\t";
cin>>x>>y>>z;
int max;
if(x>y)
max=x;
if(y>x)
max=y;
if(z>x)
max=z;
cout<<"the max number is"<< max<<endl;
return0;}

```