

KINGDOM OF SAUDI ARABIA

***THE NATIONAL COMMISSION FOR ACADEMIC
ACCREDITATION & ASSESSMENT***

COURSE SPECIFICATION
HASEB 231

Revised March 2007

Course Specification

For Guidance on the completion of this template, please refer to of Handbook 2 Internal Quality Assurance Arrangements

Institution	Almajmaah University
College/Department	Al-Majma'ah Community College / Department of Natural and Applied Sciences

A Course Identification and General Information

1. Course title and code: Computers Programming 2 (Haseb 231)
2. Credit hours: 3 hours
3. Program(s) in which the course is offered. (If general elective available in many programs indicate this rather than list programs) Computer Science Major / Qualification Program
4. Name of faculty member responsible for the course Mr. Mohammad Fayez Aabed
5. Level/year at which this course is offered : 3 rd Level
6. Pre-requisites for this course (if any) Computers Programming 1 (Haseb 121)
7. Co-requisites for this course (if any) N/A
8. Location if not on main campus Community College at Al-Majma'a / Class: 2-A-2 / LAB: 6-A-1

B Objectives

1. Summary of the main learning outcomes for students enrolled in the course.

- Introducing new concepts of Computer Programming (Not Sequential one) called Object Oriented Programming that is based on the features of Encapsulation and information hiding using C++ language.
- Learning how to create and program the main feature of Object Oriented Programming which is called Class.
- Understanding the concepts of Inheritance & Polymorphism.
- Gaining the skills of catching several programming faults and errors using Exception Handling programming techniques.

2. Briefly describe any plans for developing and improving the course that are being implemented. (eg increased use of IT or web based reference material, changes in content as a result of new research in the field)

This Course is one of the frequently revised courses by committee of study's scheduling and planning to ensure its follow-up to the up-to-date development in its related major, from the following aspects:

- 1- Using up-to-date topic-related references
- 2- Revising scientific achievements that are related to course field.
- 3- Keeping track of activities for Associates and companies that are interested in course field.
- 4- Following-up outcomes from related scientific researches.
- 5- Attending Scientific conferences.

C. Course Description (Note: General description in the form to be used for the Bulletin or Handbook should be attached)

1 Topics to be Covered		
Topic	No of Weeks	Contacthours
An Introduction about basic concepts of Object oriented Programming using C++, such as: Class, Object, and Data Abstraction.	3	6 Hours /Theoretical 6 Hours /Practical
Advanced Topics in Classes and Objects using C++.	2	4 Hours /Theoretical 4 Hours /Practical
Operator Overloading	2	4 Hours /Theoretical 4 Hours /Practical
Inheritance	3	6 Hours /Theoretical 6 Hours /Practical
Polymorphism	2	4 Hours /Theoretical 4 Hours /Practical
File Processing	2	4 Hours /Theoretical 4 Hours /Practical
Revision	1	2 Hours /Theoretical 2 Hours /Practical

2. Course components (total contact hours per semester):

Lecture: 30 Hours	Tutorial:	Practical/Fieldwork /Internship: 30 Hours	Other:
-------------------	-----------	---	--------

3. Additional private study/learning hours expected for students per week. (This should be an average :for the semester not a specific requirement in each week)

N/A

4. Development of Learning Outcomes in Domains of Learning For each of the domains of learning shown below indicate:
a. Knowledge
(i) Description of the knowledge to be acquired - Reviewing basic concepts of Sequential Programming and writing codes. - Learning the concept of Object Oriented Programming (O.O.P) and its basic concepts, such as: Classes and Objects. - Learning advanced techniques in Object Oriented Scheme, like: Inheritance Polymorphism and how to apply them in programming. - Understanding File processing techniques in C++ Programming Language.
(ii) Teaching strategies to be used to develop that knowledge - In-Class Lectures - In-Lab Lectures - Outside-Learning resources - Team Discussions (Students and Teacher)
(iii) Methods of assessment of knowledge acquired - Paper-based Exams - Practical Exams - Quizes
b. Cognitive Skills
(i) Cognitive skills to be developed - Skills of using Programming Development tools (MS VS.NET 2008)

- Skills of using Internet and its tools.
- Skills of doing Search and Information retrieval over Internet
- Skills of using logical thinking in writing coding statements.
- Skills of mastering advanced programming techniques.
- Skills of analysing and developing several types of programming solutions.
- Skills of projecting learned programming concepts onto real life applications.

(ii) Teaching strategies to be used to develop these cognitive skills

Analyse, simplify, and discuss above listed skills through lectures and practical training sessions and simulate them by examples related to student's environment and several real-life applications from programming market.

(iii) Methods of assessment of students cognitive skills

- Practical lab-sessions.
- Training exercises
- Home-works and weekly assignments
- In-class Discussions between students themselves and their teacher

c. Interpersonal Skills and Responsibility

(i) Description of the interpersonal skills and capacity to carry responsibility to be developed

- The Ability to work in a team to develop a specific programming application
- The Ability to work individually to accomplish a specific task.
- The Ability to interact with other colleagues by discussions.

(ii) Teaching strategies to be used to develop these skills and abilities

- create study groups to exchange info and experiences in programming techniques between members per each group.

(iii) Methods of assessment of students interpersonal skills and capacity to carry responsibility

- Frequent evaluation for team-work projects.
- Frequent evaluation for accomplished individual assignments.

d. Communication, Information Technology and Numerical Skills

(i) Description of the skills to be developed in this domain.

<ul style="list-style-type: none"> - The Ability of using Computer system. - The Ability of using Software Development tools. - The Ability of installing software applications. - The Ability of using Internet and its tools. - The Ability of searching and retrieving info from the Internet. - The Skills of writing coding statements using programming languages. - The Skills of doing presentations and discussions with others.
<p>(ii) Teaching strategies to be used to develop these skills</p> <ul style="list-style-type: none"> - Discussions by In-Lab Applications. - Assigning several exercises. - Giving take-home assignments and home-works. - Preparing some related topics to be presented In-class by students themselves
<p>(iii) Methods of assessment of students numerical and communication skills</p> <ul style="list-style-type: none"> - Evaluating In-Lab projects. - Evaluating Exercises. - Evaluating take-home assignments and home-works. - Evaluating Student's performance by their In-class presentations and discussions.
<p>e. Psychomotor Skills (if applicable)</p>
<p>(i) Description of the psychomotor skills to be developed and the level of performance required</p> <p>N/A</p>
<p>(ii) Teaching strategies to be used to develop these skills</p> <p>N/A</p>
<p>(iii) Methods of assessment of students psychomotor skills</p> <p>N/A</p>

5. Schedule of Assessment Tasks for Students During the Semester			
Assessment	Assessment task (eg. essay, test, group project, examination etc.)	Week due	Proportion of Final Assessment
1	Assignments – Home-works – Tasks	Once per a week	10%

2	Research – Programming Applications	5 , 10	10%
3	First Exam	7	20%
4	Second Exam	12	20%
5	Final Exam	Scheduled later	40%

D. Student Support

1. Arrangements for availability of faculty for individual student consultations and academic advice.
(include amount of time faculty are available each week)
 - A Direct Supervision should be by the teacher for students at the time of In-Lab practical training.
 - On Average, 4 hours per a week for Office ones.

E. Learning Resources

1. Required Text(s) C++ How to Program H.M. & P.J. Deitel, 3rd Edition, 2001, Prentice Hall.
2. Essential References C++ Programming: From Problem Analysis to Program Design, D.S.Malik, 4th Edition, Thomson (Course Technology),2008.
3- Recommended Books and Reference Material (Journals, Reports, etc) (Attach List) C++ Unleashed , Vishwajit Aklecha, Sams, 1998.
4-.Electronic Materials, Web Sites etc Resources on the Web: http://www.kutub.info/library http://www.cplusplus.com
5- Other learning material such as computer-based programs/CD, professional standards/regulations Microsoft Visual Studio.NET 2005 software package.

F. Facilities Required

Indicate requirements for the course including size of classrooms and laboratories (ie number of seats in classrooms and laboratories, extent of computer access etc.)
1. Accommodation (Lecture rooms, laboratories, etc.) Computer Lab with 30 seats + A Lecture room with 30 seats
2. Computing resources 30 PCs
3. Other resources (specify --eg. If specific laboratory equipment is required, list requirements or attach list) - Shared Laser Printer. - Microsoft Visual C++ software (ver. 6 or .NET) - Smart Board with a projector (at both lecture room + Computers Lab)

G Course Evaluation and Improvement Processes

1 Strategies for Obtaining Student Feedback on Effectiveness of Teaching - Evaluating the course by taking students opinions at the end of the semester from several aspects, like: teaching techniques, topics covered,...etc. - Doing some interviews with a randomly selected group of students, and asking them about their recommendations to develop the course for next semesters.
2 Other Strategies for Evaluation of Teaching by the Instructor or by the Department - An Evaluation should be made by the members of the department for the teacher of the course and the tools that were used in the teaching process. - An Evaluation from guest teachers. - Out-of-department revision. - An Evaluation by the teacher of the course himself. - A frequent local revision for the course by the committee of study's scheduling and planning.
3 Processes for Improvement of Teaching - Availability for latest up-to-date PCs and maintenance support for them. - Getting all benefits as much as we can from all in-campus and global revision processes.

- Taking in consideration all comments and recommendations made by department's committee about the teacher of the course by direct tracking for his teaching performance.

- Encourage self learning.

- Encourage extra readings from other resources.

- Encourage students doing presentations and mini-lectures.

- Encourage students interacting with each other by discussion's groups.

4. Processes for Verifying Standards of Student Achievement (eg. check marking by an independent faculty member of a sample of student work, periodic exchange and remarking of a sample of assignments with a faculty member in another institution)

- Recheck procedure for randomly selected papers from one of course exams by external examiners.

- Revision procedure for randomly selected papers from one of course exams by a special committee of teaching members of department.

- Arrange some kind of group checking procedure with all teaching members at the department.

5 Describe the planning arrangements for periodically reviewing course effectiveness and planning for improvement.

- Compare this course with similar ones at other related departments.

- update the contents of course material to contain latest up-to-date info in this field of science.

- Use new technology tools to present and teach the course.

- Getting benefits from the evaluation made by students to add, modify, and update course material and teaching methodologies.

- A frequent and continuous revision should be made for the course description by the committee of study's scheduling and planning.

