

ATTACHMENT 5.

Kingdom of Saudi Arabia
The National Commission for Academic Accreditation &
Assessment

14011101-4 Computer Programming
(CS)

Course Specifications

Institution	Umm Al Qura University	Date	7 / 7 / 1437
College/Department	College of Computers and Information Systems		

A. Course Identification and General Information

1. Course title and code: 14011101-4 Computer Programming			
2. Credit hours 4			
3. Program(s) in which the course is offered. Computer Science			
4. Name of faculty member responsible for the course Curriculum Committee			
5. Level/year at which this course is offered 2nd year / level 3			
6. Pre-requisites for this course 4800153-3 Computer Programming Skills			
7. Co-requisites for this course			
8. Location if not on main campus Al-Abidiyah campus (Boys) and Al-Zaher campus (Girls), Makkah Al Mukarramah			
9. Mode of Instruction (mark all that apply)			
a. traditional classroom	<input checked="" type="checkbox"/>	What percentage?	<input type="text" value="100"/>
b. blended (traditional and online)	<input type="checkbox"/>	What percentage?	<input type="text"/>
c. e-learning	<input type="checkbox"/>	What percentage?	<input type="text"/>
d. correspondence	<input type="checkbox"/>	What percentage?	<input type="text"/>
f. other	<input type="checkbox"/>	What percentage?	<input type="text"/>
Comments:			

B Objectives

1. What is the main purpose for this course?

This course introduces students to the basics of writing software programs using the Java programming language. It focuses on how to use various Java languages constructs to develop high quality, working code that solves real problems. Topics covered include the Java programming language syntax, variables, data types, control structures, Input/output, methods, arrays, classes, objects, and general rules for writing good code. Students will develop and test Java programs (~~typically~~) using NetBeans a commercial IDE. In this course, the student will gain the ability to:

1. Analyze problems and develop computer algorithms to solve ~~novel~~ simple problems.
2. Write, document, test, and debug Java programs, making use of variables, expressions, selection and looping statements.
3. Use of professional programming coding style and comments to improve code readability and maintainability.
4. Organize program code into modules using methods following the software engineering principles of modularity and abstraction.
5. Assemble data and methods into classes at an introductory level following the software engineering principles of encapsulation and data hiding.
6. Make use of arrays to store and process lists of data.
7. Read, interpret, analyze, and explain introductory Java programs.

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

1. Increase the use of the latest Web-based reference material and textbooks.
2. Review and update the course materials as part of preparation to teach this course.
3. Gather students' opinions about their success in achieving course objectives by the end of the semester. This is done through number of survey questions that map one-to-one with course objectives.
4. Review and indicate which assessment instrument(s) to be used for assessing each course outcome, and what grading rubric will be used for each instrument.

C. Course Description (Note: General description in the form used in Bulletin or handbook)

Course Description:

1. Topics to be Covered

List of Topics	No. of Weeks	Contact hours
Introduction (history/features of Java, how Java works, types of Java programs)	1	3
Edit, compile, debug, and run Java Programs	1	3
Variables, data types, assignment, expressions, and programming style	2	3
<u>Methods (arguments/parameters, call by-value/reference, overloading, local/field variables, static class members, and return values)</u>	<u>2</u>	<u>3</u>
Input/output via console and dialog boxes	1	3
Selection structure (if, if-else, and switch-case)	2	3
Repetition structure (for, while, and do-while)	2	3
Arrays (one-dimensional and multidimensional)	2	3
<u>Methods (arguments/parameters, call by-value/reference, overloading, local/field variables, static class members, and return values)</u>	<u>2</u>	<u>3</u>
Object-based programming (classes/objects, instance class members, member access modifiers, constructors, set/get methods, and this reference)	2	3

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

	Lecture	Tutorial	Laboratory or Studio	Practical	Other:	Total
Contact Hours	30	15	42			
Credit	<u>???</u>	<u>???</u>	<u>???</u>			

3. Additional private study/learning hours expected for students per week.

<u>6 hours</u>

4. Course Learning Outcomes in NQF Domains of Learning and Alignment with Assessment Methods and Teaching Strategy

On the table below are the five NQF Learning Domains, numbered in the left column.

First, insert the suitable and measurable course learning outcomes required in the appropriate learning domains (see suggestions below the table). **Second**, insert supporting teaching strategies that fit and align with the assessment methods and intended learning outcomes. **Third**, insert appropriate assessment methods that accurately measure and evaluate the learning outcome. Each course learning outcomes, assessment method, and teaching strategy ought to reasonably fit and flow together as an integrated learning and teaching process. (Courses are not required to include learning outcomes from each domain.)

Code	NQF Learning Domains	Course Teaching	Course Assessment
------	----------------------	-----------------	-------------------

#	And Course Learning Outcomes	Strategies	Methods
1.0	Knowledge		
1.1	Basic understanding of the Java programming language	Lectures, tutorial, labs, assignments	Quiz, lab evaluation, projects, assignment evaluation
1.2	Gain the ability to design simple programs	Lectures, tutorial, labs, assignments	Quiz, lab evaluation, projects, assignment evaluation
1.3	Understand and use basic control structures	Lectures, tutorial, labs, assignments	Quiz, lab evaluation, projects, assignment evaluation
1.4	Understand and use arrays, Vectors and Hash tables	Lectures, tutorial, labs, assignments	Quiz, lab evaluation, projects, assignment evaluation
1.5	Gain the ability to design basic objects	Lectures, tutorial, labs, assignments	Quiz, lab evaluation, projects, assignment evaluation
2.0	Cognitive Skills		
2.1	Will acquire the ability to understand basic algorithm	Lectures, tutorial, labs	Quiz, lab evaluation, projects
2.2	Will acquire the ability to design basic algorithm	Lectures, tutorial, labs	Quiz, lab evaluation, projects
3.0	Interpersonal Skills & Responsibility		
3.1			
3.2			
4.0	Communication, Information Technology, Numerical		
4.1	Improve the ability to work in a group	Projects	Project evaluation
4.2			
5.0	Psychomotor		
5.1	Demonstrate skills in using computer machines and software tools to solve computer problems	Lab exercise, lab demonstration	Lab. exams In-lab. evaluation
5.2	Perform a task with minimum assistance	Lab exercise	Lab. exams In-lab. evaluation

5. Map course LOs with the program LOs. (Place course LO #s in the left column and program LO #s across the top.) (I = Introduction P = Proficient A = Advanced)

Course LOs #	Program Learning Outcomes (Use Program LO Code #s provided in the Program Specifications)														
	1.1	1.2	2.1	2.2	2.3	2.4	2.5	3.1	3.2	3.3	4.1	4.2	4.3	5.1	5.2
1.1			I	I	I	I	I			I		I	I		
1.2			I	I	I	I	I			I		I	I		
1.3			I	I	I	I	I			I		I	I		
1.4			I	I	I	I	I			I		I	I		
1.5			I	I	I	I	I			I		I	I		
2.1			I	I	I	I	I			I		I	I		
2.2			I	I	I	I	I			I		I	I		
4.1								I		I	I				
5.1										I				I	I
5.2										I				I	I

6. Schedule of Assessment Tasks for Students During the Semester

	Assessment task (e.g. essay, test, group project, examination, speech, oral presentation, etc.)	Week Due	Proportion of Total Assessment
1	Assignments	-	0
2	Quiz 1	5	10
3	Midterm	8	25
4	Quiz 2 Project	11	10
5	Lab Exam	15	15
6	Final exam	16	40
7			
8			

D. Student Academic Counseling and Support

1. Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice. (include amount of time teaching staff are expected to be available each week)

Office hours between 2-4 hours per week.

E Learning Resources

1. List Required Textbooks

Java: How to Program, 9e, Dietel and Dietel, Pearson 0273759760

2. List Essential References Materials (Journals, Reports, etc.)

Lecture slides and notes

3. List Recommended Textbooks and Reference Material (Journals, Reports, etc)

Absolute Java, 4th Edition, Walter Savitch, Addison Wesley, 2009

Java Programming, 7th Edition, Joyce Farrell, 2013, 1285081951

4. List Electronic Materials, Web Sites, Facebook, Twitter, etc.

The Java Tutorials: <http://docs.oracle.com/javase/tutorial/getStarted/index.html>

Language Basics: <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>

Java for Complete Beginners: <http://www.homeandlearn.co.uk/java/java.html>

5. Other learning material such as computer-based programs/CD, professional standards or regulations and software.

The main textbook: Java example codes

F. Facilities Required

Indicate requirements for the course including size of classrooms and laboratories (i.e. number of seats in classrooms and laboratories, extent of computer access etc.)

1. Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)

Lecture room (max 40 students)

Computer lab (max 20 students)

Overhead projector and internet connection
2. Computing resources (AV, data show, Smart Board, software, etc.) Integrated Development Environment (e.g., NetBeans, Eclipse, JBuilder). Java Development Kit (JDK)
3. Other resources (specify, e.g. if specific laboratory equipment is required, list requirements or attach list)

G Course Evaluation and Improvement Processes

1 Strategies for Obtaining Student Feedback on Effectiveness of Teaching A student-feedback form is distributed at the end of the course.
2 Other Strategies for Evaluation of Teaching by the Instructor or by the Department
3 Processes for Improvement of Teaching
4. Processes for Verifying Standards of Student Achievement (e.g. check marking by an independent member teaching staff of a sample of student work, periodic exchange and remarking of tests or a sample of assignments with staff at another institution)
5 Describe the planning arrangements for periodically reviewing course effectiveness and planning for improvement.

Name of Instructor: _____

Signature: _____ Date Report Completed: _____

Name of Course Instructor _____

Program Coordinator: _____

Signature: _____ Date Received: _____