# Course Specifications

**Programme(s) on which the course is given** B.Sc. of Pure Mathematics and Computer Science, Physics and computer science

**Major or minor element of programs** Major
**Department offering the program** Mathematics
**Department offering the course** Mathematics
**Academic year / Level** Second level (2)
**Semester**
**Date of specification revision** September 2012
**Date of specification approval** September 2012

## A- Basic Information

**Title:** Computer Language **Code:** M2311

**Credit Hours:** 4 **Total:** 4 hr.

**Lecture:** 3 **Tutorial:** - **Practical:** 2 **Other:** -

## B- Professional Information

### 1 – Overall Aims of Course

The students learn the principles of object oriented programming including classes, polymorphism, encapsulation and information hiding, and inheritance. Principles of object oriented design. Program debugging and documentation techniques. Implementation and simple analysis of algorithms for sorting and searching. Event-driven programming and the use of libraries for user interfaces. Introduction to computer history. Programming assignments in JAVA.

### 2 – Intended Learning Outcomes of Course (ILOs)

**a- Knowledge and Understanding:**

**The student should be able to**

**a1- Understand the knowledge in designing and implementing an Object oriented program**

**a2- Recognize program structure baring in mind trade.**

**a3- Demonstrate how to document the program using Javadoc and**

**a4- Demonstrate how to debug using IDEs debugging tools.**

**b- Intellectual Skills**

**b1- Construct objects and their behavior.**

**b2- Apply programming with technical and professional programming style to design small projects.**

**b3- Develop and evaluate the docs of any programming language for future upgrades.**

**c- Professional and Practical Skills**

**c1- Choose and apply classes with polymorphism.**

**c2- Develop skills of creating graphical user interface.**

**c3- Develop skills in writing html comments for the documentation process.**

**d- General and Transferable Skills**

**d1- Work in a team design object oriented projects**

**d2- Exhibit appropriate the right style in programs development**

**d3- Think independently for testing programs through the debugging process for producing reliable programs.**

**3- Contents**

| Topics | No. of hours | Lecture | Tutorial/Practical |
|---|---|---|---|
| **Introduction**<br>• **History of** | **4** | **1** | **1** |

| | | | |
|---|---|---|---|
| **programming languages**<br>• **Algorithms**<br>• **Implementing Algorithms in Java**<br>• **Why Java?**<br>• **The Web and GUIs** | | | |
| **Program Fundamentals**<br>• **Compiling and Running Java**<br>• **Lexical Elements**<br>• **Primitive Data Types and Variables**<br>• **User Input**<br>• **Predefined Methods**<br>• **Number Types**<br>• **Arithmetic Expressions**<br>• **Assignment Operators**<br>• **Increment and Decrement Ops**<br>• **Precedence and Associatively**<br>• **Basic elements of a program (javadoc comments)** | **8** | **2** | **2** |
| **Statements and Control Flow** | **12** | **3** | **3** |

| | | | |
|---|---|---|---|
| • **Kinds of Statements**<br>• **Boolean Expressions**<br>• **The if Statement (and if-else)**<br>• **The while Statement**<br>• **The for Statement**<br>• **Break and Continue**<br>• **The switch Statement**<br>• **Laws of Boolean Algebra** | | | |
| **Methods: Functional Abstraction**<br>• **Method Invocation**<br>• **Static Expressions**<br>• **Scope of Variables**<br>• **Top-Down Design**<br>• **Invocation and Call-by-value**<br>• **Recursion**<br>• **Method Overloading**<br>• **Packages as String, Math, etc.** | **12** | **3** | **3** |

| | | | |
|---|---|---|---|
| **5. Arrays**<br>• **One-Dimensional Arrays**<br>• **Passing Arrays to Methods**<br>• **Array Alignment**<br>• **Finding Min and Max**<br>• **Simple Sorting**<br>• **Searching an Ordered Array**<br>• **Two-Dimensional Arrays**<br>• **Arrays of Nonprimitive Types** | **9** | **2** | **3** |
| **6. Objects: Data Abstraction**<br>• **String: Using a Standard Class**<br>• **StringBuffer**<br>• **Elements of a Simple Class**<br>• **Access public and private: Data Hiding**<br>• **Constructor Methods**<br>• **Static Fields and Methods**<br>• **Passing Objects:** | **12** | **3** | **3** |

| | | | |
|---|---|---|---|
| **Reference Types**<br>• **Scope**<br>• **Keyword final and constants**<br>• **Arrays of Objects**<br>• **Object-oriented Design** | | | |

**4– Teaching and learning methods**

**4.1- Lectures**

**4.2- Working on hand in assignments**

**4.3- Attending practical classes**

**5- Student assessment methods**

**5.1 Mid term written exam to assess understanding competencies**

**5.2 Mid term practical Exam to assess programming skills**

**5.3 Oral Exam to assess attendance and interesting.**

**5.4 Semester hand in assignments to assess understanding professionalism.**

**5.5 Final lab exam to assess a whole lab skills**

**5.6 Final term written Exam to assess comprehension.**

**Assessment schedule**

| | | |
|---|---|---|
| **Assessment 1** | **Mid term + practical** | **Week 7** |
| **Assessment 2** | **semester activities** | **Week 5 and 8** |
| **Assessment 3** | **Final term oral exam + lab** | **Week 13** |
| **Assessment 4** | **final term written exam** | **Week 14** |

**Weighting of assessments**

**Mid-Term Examination**

**20%**

**Semester Work (homework assignments + lab + oral tests)**
**20 %**
**Other types of assessment**
**00%**
**Final-term written Examination**
**60%**
**Total**
**100%**

**Any formative only assessments**

**6- List of references**

**6.1- Course notes**

**Collected and prepared notes that covers the main topics in the course content**

**6.2- Essential books (text books)**

**Elementary textbooks under the title: *Java Foundations, Lowis, DePasquale and Chase. Addison Welsey 2007***

**6.3- Recommended books:**

- **Java an introduction to computer science and programming. Savitch. Printice Hall 1999**
- **Objects first with Java. Barnes and Kolling Pearson. 3ed. 2006**
- ***Java: How to program***

**6.4- Periodicals, Web sites, … etc**

**Non.**

**7- Facilities required for teaching and learning**

**Lecture: PC's - packages for ready-made scientific programs. - Data Show**

**Lab: Advanced lab contains all the network tools, instrumentation, and packages.**

**Course coordinator: Dr. Passent El-Kafrawy**

**Head of Department: Prof. Dr. Mohamed A. Ramadan**

**Date: / /**