SCICOMP102, Introduction to Computer Science
[Fall 2019]

[SCICOMP102; Introduction to Computer Science]

[Fall 2019]

Classroom no: **F-18**
Class times: **MON 16:00-18:00 THU 16:00-18:00**

Instructor: **Dr. Andrew Brooks**
Email: **a.brooks@ucr.nl**
Tel: **0118 655 534**
Office no. & location: **Eleanor 1.09**
Office hours: **appointments - see Moodle scheduler activity or walk-in**

I. Track information

a) Prerequisites for this course: **none**
b) This course serves as a prerequisite for all other computer science courses.

This course is part of the Computer Science track. For further information about the track, please see the track document available on the UCR intranet.

II. Course description

This is an introductory course in problem solving and computer programming in Java. Although Java is an object oriented programming language, the course begins by introducing traditional structured programming and data constructs (i.e. selections, loops, methods, primitive types, and arrays). Then consideration is given to the object-oriented programming constructs (i.e. encapsulation, composition, inheritance, polymorphism, abstract classes, and interfaces). The second meeting of the class each week is entirely devoted to laboratory work where students tackle programming exercises and demonstrate their work. Homework comprises weekly multiple-choice tests. Two larger programming projects are undertaken. There is a written midterm exam and a written final exam. By the end of the course a student will have obtained a reasonable familiarity with the Java API (Application Programming Interface) and a Java IDE (Integrated Development Environment).

III. Study Load

This course earns students four credits (equivalent to 7.5 ECTS). The class meets twice a week for two hours. Preparation time is approximately 10 hours per week.

IV. Course materials

The following textbook is required for this course:

Introduction to Java Programming, Brief Version, 11th Edition
By Y. Daniel Liang, Published by Pearson Education Limited © 2019
Global Edition
ISBN-10: 1292222034 ISBN-13: 9781292222035

V. Course organization and requirements

The first session of each week will cover content from the required textbook. This session will involve a mix of lecturing, discussion-based learning, and various other in-class activities. The second session of each week will be a laboratory in which programming exercises are undertaken. The instructor will provide assistance where needed and sign out completed exercises when they have been successfully demonstrated.

Homework and laboratories will be issued approximately on a weekly basis. If the number of issued homework or laboratories exceeds 10, the additional work will count as extra-credit. Other extra-credit opportunities may be available at the instructor's discretion.

Students are expected to:

(i) bring the required textbook to every class
(ii) bring their own fully-charged laptops, if they have them, to every class, for programming
(iii) read relevant book chapter(s) and make full use of the book's learning resources such as videos and sample exercise solutions
(iv) participate actively in class when asked to do so
(v) monitor Moodle, the official mode of communication, on a daily basis
(vi) attend class and inform the instructor beforehand if they cannot attend because of illness or some other urgent reason
(vii) work consistently on the laboratory exercises outside of class meetings
(viii) tackle the laboratory exercises and programming projects individually
(ix) be able to explain the programming code they have written
(x) not use mobile devices in class other than their own laptops for programming
(xi) not redistribute materials made available in Moodle to third parties
(xii) attend office hours for help and guidance on any aspect of the course when required

Homework deadlines are firm. Homework should be completed one week from the date of issue. Laboratories have full value for one week from the date of issue. After one week, their value is halved. After two weeks, they no longer have value.

This course is subject to UCR academic rules and procedures. Both students and instructors are required to know and follow these rules and procedures. Students should not commit acts of plagiarism. Students are advised that if they miss more than 6 class sessions they will receive an automatic F (**i.e. 6 absences only are permissible**). Two hours of lateness in attending class meetings will count as one absence.

VI. Assessment

assessed component	value	
written midterm	20%	sample questions will be provided
written final	20%	sample questions will be provided
programming project 1	10%	
programming project 2	10%	
homework (10 each at 1%)	10%	
laboratories (10 each at 3%)	30%	

Homework, laboratories, written midterm and written final will be assessed on correctness of answers. Partial credit will be awarded for partial correctness.

Comprehension questions will be asked of laboratory work. Partial credit will be awarded when comprehension questions are not fully answered.

The written midterm and written final may incorporate wild card questions based on homework (up to 2% of the 20%).

Projects are assessed on a sliding scale of accomplishment. Full marks are awarded only if all the project requirements are met, all test cases pass and all comprehension questions are fully answered. **Failure to answer a project comprehension question attracts a penalty of 1%.**

Full project specifications are available in Moodle.

VII. Course schedule

The course schedule may be subject to change. For example, if things are progressing very well, it might be possible to occasionally tackle more material in a week. National holidays or special college events may result in some content being covered in less depth. Key dates within any week will be posted in Moodle

Time	**Topics to be discussed**	**Course material used**	**Assignments and assessment**
Week 1 [26 Aug]	Introduction to Computers, Programs, and Java. Elementary Programming.	Chapters 1 & 2	homework & laboratory
Week 2 [2 Sep]	Selections. Mathematical Functions, Characters, and Strings.	Chapters 3 & 4	homework & laboratory
Week 3 [9 Sep]	Loops.	Chapter 5	homework & laboratory
Week 4 [16 Sep]	Methods.	Chapter 6	homework & laboratory ~ project 1 issued
Week 5 [23 Sep]	Single-Dimensional Arrays. Multidimensional Arrays.	Chapter 7 & 8	homework & laboratory
Week 6 [30 Sep]	continuation of Week 5 material	Chapter 7 & 8	homework & laboratory
Week 7 [7 Oct]	Objects and Classes.	Chapter 9	homework & laboratory
BREAK [14 Oct]			
Week 8 [21 Oct]			**midterm exam (20%) project 1 deadline (10%)**
Week 9 [28 Oct]	Object-Oriented Thinking.	Chapter 10	homework & laboratory
Week 10 [4 Nov]	Inheritance and Polymorphism.	Chapter 11	homework & laboratory
Week 11 [11 Nov]	Exception Handling and Text I/O.	Chapter 12	homework & laboratory ~ project 2 issued
Week 12 [18 Nov]	Abstract Classes and Interfaces.	Chapter 13	homework & laboratory
Week 13 [25 Nov]	Java FX Basics.	Chapter 14	homework & laboratory
Week 14 [2 Dec]	Event-Driven Programming and Animations	Chapter 15	
Week 15 [9 Dec]			**final exam (20%) project 2 deadline (10%)**

VIII. Student learning outcomes

Upon successfully completing this course, a student should be able to:

SLO 1 code, trace the execution of, and debug non-GUI applications, comprising several Java classes, which use traditional structured programming and data constructs (i.e. selections, loops, methods, primitive types, and arrays)

SLO 2 demonstrate a knowledge and understanding of traditional structured programming and data constructs

SLO 3 code, trace the execution of, and debug non-GUI applications, comprising several Java classes, which use object-oriented programming constructs (i.e. encapsulation, composition, inheritance, polymorphism, abstract classes, and interfaces) and exception handling

SLO 4 demonstrate a knowledge and understanding of object-oriented programming constructs and exception handling

SLO 5 solve problems of moderate complexity by writing computer programs in Java

SLO 6 demonstrate a knowledge and understanding of simple GUI applications, comprising several Java classes, which make use of events, event handlers, and simple GUI components (e.g. labels, text fields, and buttons)

Period	**Teaching activities**	**Student is able to do**
Weeks 1-8	lectures & homework & laboratories	SLO 1
Week 8	written midterm	SLO 2
Weeks 9-15	lectures & homework & laboratories	SLO 3
Week 15	written final	SLO 4
Weeks 4-8 & Weeks 11-15	Project 1 and Project 2	SLO 5
Weeks 14-15	lectures & written final	SLO 6