

## COP2800

## Computer Programming Using JAVA

Fall 2019 - On Campus

**CLASS LOCATION:** CSE 220.

**CLASS MEETING TIME(S):**

- Tuesdays | Period 7 (1:55 PM - 2:45 PM)
- Thursdays | Periods 7 - 8 (1:55 PM - 3:50 PM)

**INSTRUCTOR:** Mr. Pedro Guillermo Feijóo-García, M.Sc.

- a. Office location: CSE E451.
- b. Email address: [pfeijoogarcia@ufl.edu](mailto:pfeijoogarcia@ufl.edu) (please use Canvas email to contact me).

**OFFICE HOURS:** TBD.

**COURSE TA'S:** TBD.

**COURSE WEBSITE:** available on <https://elearning.ufl.edu/>

**COURSE COMMUNICATIONS:** We will have a Slack channel for the course. Questions and Answer will be provided through Slack (mainly) and Canvas (Email and Q/A Forum).

**REQUIRED TEXTBOOK:** No textbook is required for this course. Readings in the form of textbook chapters and online resources will be posted to the course website. Students will be responsible for accessing the readings and downloading any relevant links provided.

**COURSE DESCRIPTION:** *Computer Programming Using JAVA* is an introductory course in computer programming that presents the process of searching for alternatives to transform situations, by designing programs using the Object-Oriented Programming (OOP) paradigm. Using JAVA as the main programming language of the course, students will learn how to use the computer as a medium to transform the world and its different contexts. The course presents concepts of Computer Science and the process of computer programming, approaching them with the Object-Oriented Programming (OOP) paradigm.

**CORREQUISITE KNOWLEDGE AND SKILLS:** MAC 1147 or equivalent.

**PURPOSE OF COURSE:** The permanent evolution and the use of information and communication technologies (ITCs) as present and future support of information

systems, require individuals with skills to analyze, design and build computer programs that offer alternatives to transform situations in different areas of knowledge.

*Computer Programming Using JAVA* is an introductory course in computer programming that presents the process of searching for alternatives to transform situations, by designing programs using the Object-Oriented Programming (OOP) paradigm. Using JAVA as the main programming language of the course, students will learn how to use the computer as a medium to transform the world and its different contexts.

**LEARNING OBJECTIVES:** By the end of this course, students will:

- be able to understand the principles of the Object-Oriented Programming (OOP) paradigm, such as: object, class, attribute, method, association.
- be able to design, develop and test simple stand-alone programs using a programming language (JAVA).
- be able to model, at a basic level, real-life situations using computational thinking and systems' modeling (UML).
- be able to differentiate between situations that may or may not be computable.
- be able to describe what programming is.

**GENERAL CONTENT:**

1. Introduction to the Object-Oriented Programming (OOP) paradigm: Classes, objects, attributes, associations and methods.
2. Computers: Fundamentals, programming languages and the compiler's function.
3. Data types: Primitive and composite.
4. Constants and Variables.
5. Identification of requirements and solution design.
6. Solution design – UML class diagrams, pseudo-code, flow charts.
7. Arithmetic, logic and assignment expressions.
8. Predicates, conditionals and the conditional cascade.
9. Linear data structures of fixed and dynamic size.
10. Bi-dimensional data structures of fixed size.
11. Control instructions and loops: for, while, do-while.
12. Searching and sorting techniques.
13. Error handling: Exception, Try-catch.
14. Basic File I/O: Properties & Text files.

15. User interaction: Console I/O.
16. User interaction: Design and development of standalone Graphical User Interfaces (GUIs).
17. OOP: Introduction to inheritance & polymorphism.

**\*Disclaimer:** Topics will not be approached in the order presented, and the depth corresponding to each one of them may vary based in the course and the students' learning processes.

**INSTRUCTIONAL METHODS:** Weekly meetings in lecture format, video tutorials and lectures, student-created artifacts (video-tutorials and mental models), homework assignments, projects, and exams. Also, we will use co-evaluations strategies throughout the course, which are proposed to enhance students' learning outcomes.

**TENTATIVE CALENDAR - SUBJECT TO CHANGE\***

The course will be organized in three modules:

Module	Dates	Topics
<p><b>First Module</b> *Weekly progress will depend on the classroom and its progress.</p>	<p><b>(08/19 - 09/30)</b></p>	<ul style="list-style-type: none"> <li>• Introduction to the Object-Oriented Programming (OOP) paradigm: Classes, objects, attributes, associations and methods.</li> <li>• Computers: Fundamentals, programming languages and the compiler's function.</li> <li>• Data types: Primitive and composite.</li> <li>• Constants and Variables.</li> <li>• Identification of requirements and solution design.</li> <li>• Solution design - UML class diagrams, pseudo-code, flow charts.</li> <li>• Arithmetic, logic and assignment expressions.</li> <li>• Predicates, conditionals and the conditional cascade.</li> </ul>
<p><b>Second Module</b> *Weekly progress will depend on the classroom and its progress.</p>	<p><b>(10/01 - 11/04)</b></p>	<ul style="list-style-type: none"> <li>• Linear data structures of fixed and dynamic size.</li> <li>• Control instructions and loops: for, while, do-while.</li> <li>• Searching and sorting techniques.</li> </ul>

		<ul style="list-style-type: none"> <li>• Error handling: Exception, Trycatch.</li> </ul>
<p><b>Third Module</b> *Weekly progress will depend on the classroom and its progress.</p>	<p><b>(11/05 - 12/11)</b></p>	<ul style="list-style-type: none"> <li>• Bi-dimensional data structures of fixed size.</li> <li>• User interaction: Design and development of standalone Graphical User Interfaces (GUIs).</li> <li>• OOP: Introduction to inheritance &amp; polymorphism.</li> <li>• <i>Basic File I/O: Properties &amp; Text files.</i></li> </ul>

### RECOMMENDED LITERATURE:

#### \*Literature in English\*

1. Kurniawan, B. (2010). *Java: A beginner's tutorial (2nd edition)*. Retrieved from <https://ebookcentral.proquest.com/lib/ufl/detail.action?docID=3003873>
2. Schildt, H. (2019). *Java: A Beginner's Guide (8th edition)*. McGraw-Hill Education.
3. Liang, Y. D. (2018). *Introduction to Java programming and data structures*. New York, NY: Pearson Education.
4. Horstmann, C. S. (2015). *Big Java*. New York: Wiley.

#### \*Some useful literature that is in Spanish\*

5. Villalobos Salcedo, J. A., Casallas Gutiérrez, R., & Castillo, M. F. (2015). *Fundamentos de programación: Aprendizaje activo basado en casos*. México: Pearson Educación. Retrieved from <https://universidad-de-losandes.gitbooks.io/fundamentos-de-programacion/content/>
6. Joyanes Aguilar, L., & Zahonero Martínez, I. (2011). *Programación en JAVA: Algoritmos, programación orientada a objetos e interfaz gráfica de usuario*. México; Bogotá; Buenos Aires: McGraw Hill.
7. Deitel, H. M., Deitel, P. J., & Romero Elizondo, A. V. (2004). *Cómo programar en java (5a ed.)*. México: Pearson y Prentice Hall.

## **COURSE POLICIES:**

**MAKE-UP POLICY:** Consistent with university policies that can be found at: <https://catalog.ufl.edu/ugrad/current/regulations/info/attendance.aspx>

**ASSIGNMENT POLICY: Assignments will be due according to activities' deadlines.** No late assignment will be accepted. All re-grade requests must be made within one week of the day the activity is graded and returned.

**FINAL EXAM:** There will not be a final exam.

**EXTRA CREDIT:** Opportunities to earn extra credit are not promised, but the instructor may opt to offer supplemental assignments for extra credit. Details will be determined per assignment and must be agreed upon by both the instructor and the student.

**COURSE TECHNOLOGY:** This course will be conducted on the Canvas Learning Management System. We will work using **JAVA** as programming language, and **Eclipse** as Integrated Development Environment (IDE). Technologies' Links and tutorials will be provided through Canvas as the course flows and they are needed.

## **UF POLICIES:**

### **UNIVERSITY POLICY ON ACCOMMODATING STUDENTS WITH DISABILITIES:**

Students requesting accommodation for disabilities must first register with the Dean of Students Office (<http://www.dso.ufl.edu/drc/>). The Dean of Students Office will provide documentation to the student who must then provide this documentation to the instructor when requesting accommodation. You must submit this documentation prior to submitting assignments or taking the quizzes or exams. Accommodations are not retroactive; therefore, students should contact the office as soon as possible in the term for which they are seeking accommodations.

**UNIVERSITY POLICY ON ACADEMIC MISCONDUCT:** Academic honesty and integrity are fundamental values of the University community. Students should be sure that they understand the UF Student Honor Code at <https://www.dso.ufl.edu/sccr/process/student-conduct-honor-code/>

**\*\*NETIQUETTE: COMMUNICATION COURTESY:** All members of the class are expected to follow rules of common courtesy in all email messages, threaded discussions and chats. [Describe what is expected and what will occur as a result of

improper behavior]

<http://teach.ufl.edu/wpcontent/uploads/2012/08/NetiquetteGuideforOnlineCourses.pdf>

## GETTING HELP:

For issues with technical difficulties for E-learning in Canvas, please contact the UF Computing Help Desk at:

- [helpdesk@ufl.edu](mailto:helpdesk@ufl.edu)
- (352) 392-HELP - select option 2
- <https://request.it.ufl.edu>

\*\* Any requests for make-ups due to technical issues MUST be accompanied by the ticket number received from LSS when the problem was reported to them. The ticket number will document the time and date of the problem. You MUST e-mail your instructor within 24 hours of the technical difficulty if you wish to request a make-up.

Other resources are available at <http://www.distance.ufl.edu/getting-help> for:

- Counseling and Wellness resources
- Disability resources
- Resources for handling student concerns and complaints
- Library Help Desk support

Should you have any complaints with your experience in this course please visit <http://www.distance.ufl.edu/student-complaints> to submit a complaint.

## GRADING POLICIES:

Module	Dates	Topics
<b>First Module (30%)</b>	<b>(08/19 – 09/30)</b>	<ul style="list-style-type: none"><li>• Quizzes, homework and/or assignments: <b>10%</b></li><li>• Theoretical Evaluation: <b>5%</b></li><li>• Practical Evaluation: <b>15%</b></li></ul>

<b>Second Module (30%)</b>	<b>(10/01 – 11/04)</b>	<ul style="list-style-type: none"> <li>• Quizzes, homework and/or assignments: <b>10%</b></li> <li>• Theoretical Evaluation: <b>5%</b></li> <li>• Practical Evaluation: <b>15%</b></li> </ul>
<b>Third Module (40%)</b>	<b>(11/05 – 12/11)</b>	<ul style="list-style-type: none"> <li>• Quizzes, homework and/or assignments: <b>10%</b></li> <li>• Theoretical Evaluation: <b>15%</b></li> <li>• Practical Evaluation: <b>15%</b></li> </ul>

**\*Disclaimer:** The number and type of activities per module may vary according to the instructor's criteria.

**GRADING SCALE:** A(100-93), A-(92-90), B+(89-87), B(86-83), B-(82-80), C+(79-77), C(76-73), C-(72-70), D+(69-67), D(66-63), D-(62-60), F(59-0)

**\*Grades might be curved\***

“A ‘C’ will not be a qualifying grade for critical tracking courses. In order to graduate, students must have an overall GPA and an upper-division GPA of 2.0 or better (C or better). Note: a C- average is equivalent to a GPA of 1.67, and therefore, it does not satisfy this graduation requirement. For more information on grades and grading policies, please visit:

<https://catalog.ufl.edu/ugrad/current/regulations/info/grades.asp>”

**\*Disclaimer:** This syllabus represents my current plans and objectives. As we go through the semester, those plans may need to change to enhance the class learning opportunity. Such changes, communicated clearly, are not unusual and should be expected.