

**SECTION A-IV: NEW COURSE**  
**Department of Computer and Information Science**

Date of departmental approval: April 12, 2016

Effective date: Fall, 2017

First offering: Fall, 2017

**CISC 1115 Introduction to Programming Using Java**  
**3 hours lecture, 2 hours lab; 4 credits**

**Bulletin Description**

Algorithms, computers and programs. Writing, debugging, and testing programs. Loops and conditional control structures. Method definition and parameter passing. Arrays, ArrayLists, and Strings. Sorting, searching and other basic algorithms. Input and output. Programming applications selected from various disciplines. History and basic concepts of computer science.

**Prerequisite:** none

**Contact hours:** 5

**Frequency of Offering:** every semester, winter and summer session

**Projected enrollment:** 30 sections of 30 students annually

**Clearances:** None

**Rationale:** This course is part of a shift from using the C++ programming language to the Java programming language in the early courses of the computer science major sequence. Java offers several advantages over C++. Because it was not developed with the constraint of backward compatibility with a large existing code base, it avoids many of the pedagogically troubling inconsistencies and anomalies of C++. It offers a simpler model for object-oriented programming, the leading contemporary programming paradigm. It offers a more accessible path for students to develop graphical user interfaces, do network programming, database programming and graphics. Finally, starting with Java places the student in the middle of the current spectrum of programming language semantics, rather than at one extreme or the other.

**Department Goals Addressed by Course:**

1. Prepare students for the programming requirements in the computer science, information systems or multimedia majors.
2. Prepare students for occasional software development in other disciplines.

**Objectives of Course:**

1. Understand the concept of a program (i.e., a computer following a series of instructions)
2. Declare a variable with an appropriate data type (either character or numeric or boolean), initialize the variable's value, and change the value when appropriate
3. Use control structures, including a loop to repeat the execution of a series of statements and a conditional statement to select one from a series of alternatives
4. Write a method to perform a subtask and use the method as part of a larger program
5. Use arrays and ArrayLists to store multiple pieces of homogeneous data
6. Trace the execution of a program and predict the results of the execution
7. Write an algorithm for a simple task.

**Outcomes Anticipated for Course:** At the completion of this course, students will be able to:

1. Write, test and debug simple programs that use the imperative and procedural paradigms to

- solve data processing problems involving single streams of input and output.
2. Use iteration to process data stored in a collection and variable quantities of input data.
  3. Apply common iterative algorithms such as sorting, searching, and simple mathematical calculations.

**Course Outline:**

1. Basic concepts and history of computing; Basic program structure. [0.5 weeks]
2. Primitive data types; expressions and assignment; basic input/output. [1.5 weeks]
3. Control structures: loops and branches. [2.5 weeks]
4. Reference types, Strings and string processing. [1.5 weeks]
5. Calling and defining *methods*. [2 weeks]
6. Arrays and ArrayList objects. [1 week]
7. Sorting, searching, and other algorithms. [2 weeks]
8. Advanced input/output. [1 week]
9. Application programming practice. [2 weeks]
10. Final exam (cumulative).

Total 14 Weeks (excluding final exam week)

**Method of evaluation:**

Exams and Quizzes

Final Exam

Homework Assignments and Labs

**Method of assessment:** The department's general assessment methodology will be applied to this course. This involves selecting a subset of questions from the final exam and assessing the student answers in light of overall departmental learning objectives.

**Bibliography:**

Java For Everyone: Late Objects, Second edition. Cay S. Horstmann. Wiley, 2014. ISBN-13: 978-1-118-06331-6

Building Java Programs: A Back to Basics Approach, Third edition. Stuart Reges and Marty Stepp, Pearson, 2011. ISBN-10: 0136091814 ISBN-13: 978-0136091813

Effective Java, Second edition. Joshu Bloch, Addison-Wesley, 2008. ISBN-13: 860-1300201986

The Java Language Specification, Java SE 8 Edition (Java Series), First Edition, James Gosling, Bill Joy, Guy Steele, Gilad Bracha, and Alex Buckley. Addison-Wesley, 2014. ISBN-13: 978-0133900699.

<https://docs.oracle.com/javase/tutorial/>