

# VALENCIA COLLEGE

Course Syllabus Summer 2017 Hybrid



**COURSE:** COP 2800c Java Programming

**PROFESSOR:** Dr. Colin Archibald (Dr. A.)

Office: West Campus 9-140c

E-mail: [CArchibald@valenciacollege.edu](mailto:CArchibald@valenciacollege.edu)

(Please Use Blackboard messaging system)

Phone number: (407) 582-1517

Skype ID: Dr.Colin.Archibald

**Student engagement hours:** An announcement in Blackboard will show office hours. They are also posted on my office door West Campus 9-140c.

## **Catalog Description:**

The syntax and semantics of the Java programming language will be covered. Students will learn how to create Java applications with emphasis on object-oriented programming. Students will become familiar with object-oriented design, including the creation of classes in Java, and the use of existing classes as provided in the current version of the Java API.

CRN: 32070

Credit Hours: 3.0

Prerequisite(s) and Co-requisite(s): Minimum grade of C in COP 1000c

Meeting places and times: **Tuesday 6 PM – 8 PM Starting May 9, 2017**

**Also Online in Blackboard, also in Skype for office hours.**

## **EDUCATIONAL MATERIALS:**

### **REQUIRED**

**Textbook:** Introduction to Java Programming: Comprehensive. **Revel Edition.** Daniel Liang.

The development environment for assignments is Eclipse and Oracle's Java 1.8 (or later). There is no charge for these.

A **webcam and a free Skype account** are **REQUIRED** for this class.

(If you are purchasing a webcam, you might consider the Logitech model C310 as a good quality and affordable choice. The Valencia bookstore also has WebCams for about \$10.)

Online Lectures / demo available on YouTube:

<https://www.youtube.com/playlist?list=PL6F117B39113D9108>

## **ASSESSMENT METHODS AND EVALUATION:**

- **Midterm test**

This will be announced at least one week in advance. It will be held in the usual classroom and class time. It will be Closed book, and on paper.

- **12 Programming Assignments.**

Late assignments are penalized by up to 20%. Assignments more than 1 week late may not be graded. Assignments must be submitted in BlackBoard under the assignments tab (not by email attachment). The first and last Assignment must be submitted by the due date.

- **Revel Exercises, quizzes**

These exercises are done as part of the online interactive 'textbook' Each student will get a link to the online Revel-version of the textbook. This is not free, and there is no way to steal it.

- **Final Exam**

**August 1 6 PM – 8 PM in the usual classroom.**

The final exam in this course is MANDATORY. Any student not completing the exam will receive a grade of F for the course.

**\*Alternate Testing Centers:** If you are not in the region, or cannot get to the Valencia testing centers, you can make arrangements with testing centers at other colleges. You must make those arrangements on your own. It is becoming quite common to do this. Give the testing center contact information to DrA at least one week before the midterm / final exam. If this is not possible - contact DrA to attempt to make other arrangements.

### **Assignments and tests do not have the same weight in the grade calculation:**

Assignments	40%
Midterm Test	20%
Revel – online 'textbook'	10%
Final Exam	30%

The sum of these will determine a letter grade as follows:

90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
0 – 59	F

---

## **ATTENDANCE POLICY:**

### **For Online Sections:**

- Attendance is logging into Blackboard, reading and posting on the discussion boards, and submitting the assignments. When more than 3 assignments are past due, you will be withdrawn from the course.

### **For In-Person and Hybrid Sections:**

- It is expected that students will attend all classes. Missing more than 3 classes may result in being withdrawn from the class.
- You will lose 2% of your final course grade for each missed class (late counts as missed), up to 6% points, and then you will be withdrawn on the 4th absence for excessive absences.

It is required that students read all email and discussion postings in Blackboard.

It is expected that an average student will require a total of 8 hours of work / week to be successful in this course. **(10 hours / week in the summer semester)**

Any student who wishes to withdraw may do so in Atlas by the Withdraw date. (See **Important Dates** below)

### **NO-SHOW PROCEDURE**

Any student who does not attend class prior to the start of the no-show period for each part of term will be withdrawn by the instructor as a no-show. This will count as an attempt in the class, and students will be liable for tuition. If your plans have changed and you will not be attending this class, please withdraw yourself through your Atlas account during the drop period for this term.

**To avoid being withdrawn as a “NO-SHOW” you must attend class on May 9, or May 16. If you miss both of those classes you will be withdrawn from the class. You are still responsible for the tuition, and it counts as an ‘attempt.’ This is very harsh... don’t let it happen.**

**WITHDRAWAL** Per Valencia Policy 4-07 (Academic Progress, Course Attendance and Grades, and Withdrawals), a student who withdraws from class before the established deadline for a particular term will receive a grade of “W”. A student is not permitted to withdraw after the withdrawal deadline. A faculty member MAY withdraw a student up to the beginning of the final exam period for violation of the class attendance policy. A student who is withdrawn by faculty for violation of the class attendance policy will receive a grade of “W”. Any student who withdraws or is withdrawn from a class during a third or subsequent attempt in the same course will be assigned a grade of “F”. For a complete policy and procedure overview on Valencia Policy 4-07 please go to:

<http://valenciacollege.edu/generalcounsel/policy/>

Effective July 1, 2009, for students on Bright Futures scholarships: Students who withdraw or are withdrawn from a class must pay the college for the cost of the class (because the college must refund that cost to the State).

**MAKE-UP POLICY:** Make-up work (assignments, quizzes, exams) will be allowed in cases of documented student emergencies. For student emergencies, it is the student’s responsibility to contact the instructor and provide documentation within one week unless special arrangements have been made previously.

**ACADEMIC HONESTY:** Each student is required to follow Valencia policy regarding academic honesty. All work submitted by students is expected to be the result of the student’s individual thoughts, research, and self-expression unless the assignment specifically states “group project.” Any act of academic dishonesty will be handled in accordance with Valencia policy as set forth in the Student Handbook and Catalog.

Students who make their work available to others will also be considered to have been dishonest. Clarification: If someone passes in your homework with their name on it, you have both broken the academic honesty policy.

### **COLLEGE POLICIES:**

A full description of all College policies can be found in the College Catalog: <http://valenciacollege.edu/catalog>

Policy Manual: <http://www.valenciacollege.edu/generalcounsel/policy>

Student Handbook: <http://valenciacollege.edu/studentdev/CampusInformationServices/>

### **IMPORTANT DATES:**

Students may withdraw themselves and receive a W up until 11:59 p.m. **July 7, 2017**. Students may not withdraw themselves after that date.

**College Closed (Credit Classes Do Not Meet): May 29, July 4, 2017**

See College calendar for important dates and final exam schedule at <http://valenciacollege.edu/calendar>

### **MORE STUFF:**

*All email communication with DrA will be by Blackboard email or Skype (this is for organizational purposes).*

**STUDENTS WITH DISABILITIES:** Students with disabilities who qualify for academic accommodations must provide a Notification to Instructor (NTI) form from the Office for Students with Disabilities (OSD) and discuss specific needs with the professor, preferably during the first two weeks of class. The Office for Students with Disabilities determines accommodations based on appropriate documentation of disabilities.

East Campus Bldg. 5, Rm. 216 Ph: 407-582-2229 Fax: 407-582-8908 TTY: 407-582-1222

West Campus SSB, Rm. 102 Ph: 407-582-1523 Fax: 407-582-1326 TTY: 407-582-1222

Osceola Campus Bldg. 1, Rm. 140A Ph: 407-582-4167 Fax: 407-582-4804 TTY: 407-582-1222

Winter Park Campus Bldg. 1, Rm. 212 Ph: 407-582-6887 Fax: 407-582-6841 TTY: 407-582-1222

**STUDENT ASSISTANCE PROGRAM:** Valencia students can get immediate help with issues dealing with stress, anxiety, depression, adjustment difficulties, substance abuse, time management as well as relationship problems dealing with school, home or work. BayCare Behavioral Health Student Assistance Program (SAP) services are free to all Valencia students and available 24 hours a day by calling (800) 878-5470. Free face-to-face counseling is also available.

**DISCLAIMER:** Changes may be made at the discretion of the instructor. Any changes to this document during the semester will be delivered to each student by BlackBoard email.

**Course Content:**

- 0.1 Getting started with Eclipse. The first Java program.
1. Introduction to the Java language. The history of Java, and the reasoning that makes Java an important programming language compared to other programming languages. Java applications vs. Java applets. An introduction to Java programming relating to the internet.
  2. The eight primitive data types. The size of the eight types, numeric and boolean.
  3. The assignment operator. Arithmetic operators.
  4. An introduction to the use of existing classes - String.
  5. Console I/O
  6. Relational operators. Precedence rules for arithmetic and relational operators, including the short cut operators. Mixed data type arithmetic, including up casting, and down casting.
  7. Selection constructs. The *if* and *if – else* statements. Nested *if* statements. Logical operators, *and*, *or*, *not*.
  8. Repetition constructs. The *while*, *do – while*, and *for* loops. Nested loops. Problem solving using selection and repetition constructs.
  9. Defining a class. Instances of a class – objects. Creating methods with parameter lists and return types. Constructor methods. Instance variables and methods.
  10. Access modifiers: public private Accessors and mutators
  11. Method overloading. Advantages gained, and the beginning of the concept of polymorphism.
  12. Static methods and variables. The Math class methods.
  13. Packages and javaDoc. Packages and Access modifiers for methods and instance variables. The relationships between packages and access modifiers.
  14. Declaring and initializing arrays. Accessing array elements. Multidimensional arrays.
  15. Concepts of Inheritance. Extending existing classes. Access modifiers as they apply to inheritance. Creating super classes and base classes.
  16. Method overriding. Abstract classes and abstract methods, and their role in object-oriented design.
  17. The universal super class Object.
  18. Polymorphism. Casting objects. Passing objects to methods.
  19. Interfaces. The creation of an interface. Polymorphism using Interfaces. Constants in an interface.
- 

<b>Course Learning Outcome</b>	<b>Evidence of Learning</b>
The student will be able to describe the history and significance of the Java language within the context of the software industry.	<ul style="list-style-type: none"> <li>• Describe the history of Java, and the reasoning that makes Java an important programming language in the context of other programming languages.</li> <li>• Recognize and be able to describe the difference between Java applications and Java applets.</li> <li>• Explain the significance of Java as it applies to programming for the internet.</li> </ul>
The student will be able to code a simple Java application that uses the primitive data types, arithmetic and relational operators.	<ul style="list-style-type: none"> <li>• Define variables using any of the eight primitive data types.</li> <li>• Use variables and arithmetic operators to write statements containing arithmetic expressions.</li> <li>• Use variables and relational operators to write statements containing relational expressions.</li> <li>• Write statements that combine data types in a single expression, and promote and demote (cast) values from one primitive data type to another.</li> </ul>

<p>The student will be able to implement selection and repetition constructs in Java.</p>	<ul style="list-style-type: none"> <li>• Design and implement selections constructs using 'if' and 'if-else' statements.</li> <li>• Design and implement selection constructs using the 'switch' construct.</li> <li>• Design and implement repetition using 'while', 'do - while', and 'for' loops.</li> <li>• Design and implement nested selection and repetition.</li> </ul>
<p>The student will be able to create and use arrays.</p>	<ul style="list-style-type: none"> <li>• Declare and initialize arrays in Java.</li> <li>• Access array elements using repetition constructs.</li> <li>• Declare and initialize multi-dimensional arrays.</li> <li>• Design and code Java applications that use one- and two- dimensional arrays.</li> </ul>
<p>The student will be able to recognize and implement code that correctly uses the object-oriented principles using Java.</p>	<ul style="list-style-type: none"> <li>• Describe the concepts of class, object, method, interface, inheritance and polymorphism.</li> <li>• Invoke methods from existing classes including 'String' and 'StringBuffer.'</li> <li>• Create objects from existing classes.</li> <li>• Design and implement classes that use static variables and methods, and instance variables and methods.</li> <li>• Design and implement class hierarchies that exercise method overloading, inheritance, and polymorphism.</li> <li>• Write code that demonstrates the concepts of abstract classes, and abstract methods.</li> </ul>
<p>The student will be able to write code that uses classes that are organized in packages, and will be familiar with the most significant packages that are part of the Java API.</p>	<ul style="list-style-type: none"> <li>• Recognize the common packages in the Java API and be able to navigate through the documentation that describes the packages.</li> <li>• Implement a package of classes.</li> <li>• Implement classes that use the access modifiers public, private, protected and the default modifier.</li> </ul>
<p>The student be able to describe and implement the concept of an interface in Java, and recognize when and why to use interfaces.</p>	<ul style="list-style-type: none"> <li>• Understand polymorphism using interfaces.</li> <li>• Design and implement classes that use interfaces containing method signatures and constants.</li> </ul>
<p>The student will be able to analyze and write code that uses the try-catch-finally construct and demonstrate when it should be used according to accepted conventions.</p>	<ul style="list-style-type: none"> <li>• The student will be able to create classes that inherit from the Exception class.</li> <li>• Be able to throw and catch exceptions in a method.</li> <li>• Design and implement methods that throw exceptions that will be caught and handled in other methods.</li> <li>• Write code that appropriately uses the assert mechanism to throw AssertionError objects.</li> </ul>