



**Credit Hours:** 3

**Contact Hours:** This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

**Faculty Information:** Faculty contact information and office hours can be found on the faculty profile page.

---

## **COURSE DESCRIPTION AND OUTCOMES**

---

**Course Description:**

This course provides students with the skills needed to become a Java object-oriented programmer. Students will develop an understanding of program applications using discrete structures, access and update stored information. Students will develop an understanding of static program analysis including run-time components such as memory management in different operating system environments. Students will focus on understanding the principles of inheritance and polymorphism in object-oriented programming.

**Course Overview:**

In this course, students will focus on advanced concepts in object-oriented programming, using the Java programming language. Specifically, the programs that students develop will support typical business-processing activities. Advanced concepts related to using inheritance, polymorphism, and interfaces will be presented in the course. Students will also build applications that utilize graphical user interfaces to process user inputs and outputs and focus on advanced programming concepts, such as recursion, sorting, and the Java collections framework.

**Course Learning Outcomes:**

1. Create a program using principles of inheritance.
2. Develop a program using principles of interfaces.
3. Implement a program that utilizes a graphical user interface.
4. Utilize layout managers in the Java programming language.
5. Identify relationships between multiple classes.
6. Apply recursive methods to a program.
7. Discuss searching techniques in Java.
8. Use the collections framework in Java for a given problem.

---

## **PARTICIPATION & ATTENDANCE**

---

Prompt and consistent attendance in your online courses is essential for your success at CSU-Global Campus. Failure to verify your attendance within the first 7 days of this course may result in your withdrawal. If for some reason you would like to drop a course, please contact your advisor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

---

## COURSE MATERIALS

---

### Required:

Lysecky, R., & Lizarraga, A. (2017). *Programming in Java with ZyLabs*. Los Gatos, CA: Zyante, Inc.

The Latest Eclipse IDE (or equivalent Java development tool), Eclipse IDE for Java Developers (2018). Retrieved from <http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/>

**NOTE:** All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Please read through each course module carefully.

---

## COURSE SCHEDULE

---

### Due Dates

The Academic Week at CSU-Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT and peer responses posted by Sunday at 11:59 p.m. MT. Late posts may not be awarded points.
- **Opening Exercises:** Take the Opening Exercise before reading each week's content to see which areas you will need to focus on. You may take these exercises as many times as you need. The Opening Exercises will not affect your final grade.
- **Mastery Exercises:** Students may access and retake Mastery Exercises through the last day of class until they achieve the scores they desire.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

---

## WEEKLY READING AND ASSIGNMENT DETAILS

---

### Module 1

#### Readings

- Chapters 9 & 10 in *Programming in Java with ZyLabs*

**Opening Exercise (0 points)**

**Discussion (25 points)**

**Critical Thinking (60 points)**

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

### **Option #1: Implementing a Superclass Bank Account**

**Part 1:** Implement a superclass BankAccount that has the following fields and methods.

Fields:

- string firstName
- string lastName
- int accountID
- double balance

Methods:

- constructor(): initialize balance to zero
- deposit() – will accept a single value double parameter; the parameter value is added to the existing balance
- withdrawal() – accepts a single value double dollar amount; the parameter value is subtracted from the existing balance
- Setters and getters for firstName, lastName, and accountID
- getBalance() getter to return the balance
- accountSummary() – prints all account information

**Part 2:** Implement a CheckingAccount class that inherits from the BankAccount class, that:

- Has an interest rate attribute
- Allows overdraft withdrawals and charges a \$30 fee

Methods:

- processWithdrawal() – will display a negative balance that includes a \$30 overdraft fee and denotes that a fee has been accessed
- displayAccount() – should display all superclass attributes and provides an additional interest rate

Ensure that your program has the two required classes and a test class.

**Place each Java class into a separate Java source file.**

Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

### **Option #2: Implementing a Superclass Employee**

**Part 1:** Implement a superclass Employee that has the following fields and methods.

Fields:

- string firstName
- string lastName
- int employeeID
- double salary

Methods:

- constructor method - initialize salary field to zero.
- setters and getters for firstName, lastName, and employeeID
- employeeSummary method – prints all account attributes

**Part 2:** Implement a Manager class that inherits from the Employee class.

Fields:

- string department a

Methods:

- employeeSummary method – prints all superclass and subclass attributes

Ensure that your program has the two required classes and a test class.

**Place each Java class into a separate Java source file.**

Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

**Mastery Exercise (10 points)**

## **Module 2**

**Readings**

- Chapter 16 in Programming in Java with ZyLabs

**Opening Exercise (0 points)**

**Discussion (25 points)**

**Critical Thinking (60 points)**

Choose one of the following two assignments to complete this week. Do not do both assignments.

Identify your assignment choice in the title of your submission.

**Option #1: Creating a GUI Bank Balance Application**

Create a simple Graphical User Interface (GUI) Bank Balance application. The application must obtain the BankAccount balance from a user, and then display that balance within a JPanel when the user selects a button. The program should allow for the user to deposit and withdraw funds from their account using a simple interface. The remaining balance in the account should be displayed before exiting the program.

Ensure that your application includes the following components:

- JPanel
- JButton
- ActionListener

Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

**Option #2: Creating a GUI Application**

Create a simple GUI application that will obtain the birth date from a user. The program must then calculate and display the age of the user within a JPanel when the user selects a button. The GUI should display the value presented. Ensure that your application includes the following components:

- JPanel
- JButton
- ActionListener

**Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.**

**Mastery Exercise (10 points)**

## **Module 3**

## Readings

- Chapter 17 in *Programming in Java with ZyLabs*

## Opening Exercise (0 points)

## Discussion (25 points)

## Critical Thinking (60 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

### Option #1: Creating a User Interface I

Create a user interface that has a top bar that includes a menu. The menu should have four items.

- When the user selects the first menu option, then the date and time should be printed in a text box.
- When the user selects the second menu option, then the text box contents should be written to a text file named "log.txt."
- When the user selects the third menu item then the frame background color changes to random color hue of the color green. See [https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)
- When the user selects the fourth menu option then the program exits.

### Option #2: Creating a User Interface II

Create a user interface that has a top bar that includes a menu. The menu should have four items.

- When the user selects the first menu option, then the date and time should be printed in a text box.
- When the user selects the second menu option, then the text box contents should be written to a text file named "log.txt."
- When the user selects the third menu item then the frame background color changes to random color hue of the color orange. See [https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)
- When the user selects the fourth menu option then the program exits.

## Mastery Exercise (10 points)

## Module 4

## Readings

- Chapters 11 & 14 in *Programming in Java with ZyLabs*

## Opening Exercise (0 points)

## Discussion (25 points)

## Critical Thinking (60 points)

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

### Option #1: Creating an Abstract Class Shape I

Create an abstract class "Shape" with the following specifications:

1. An abstract method "surface\_area ()" of return type double
2. An abstract method "volume()" of return type double

Put your code in a Java source file named "Shape.java."

#### I.

Create a class "Sphere" that extends the Shape class with the following specifications:

1. Attributes:  
    Radius
2. Constructor:  
    Implement a parameterized constructor needed to initialize the data.
3. toString:  
    Implement a "toString" method that prints out the surface area and volume.

Implement methods to compute the surface area and volume.

#### II.

Create a class "Cylinder" that extends the Shape class with the following specifications:

1. Attributes:  
    Radius, height – both of type double
2. Constructor:  
    Implement a parameterized constructor to initialize the data.
3. toString:  
    Implement a "toString" method that prints out the surface area and volume.

Implement methods to compute the surface area and volume.

#### III.

Create a class "Cone" that extends the Shape class with the following specifications:

1. Attributes:  
    Radius, height – both of type double
2. Constructor:  
    Implement a parameterized constructor to initialize the data.
3. toString:  
    Implement a "toString" method that prints out the surface area and volume.

Implement methods to compute the surface area and volume.

#### IV.

Create a driver class named "ShapeArray" with the following specifications:

1. Instantiate one sphere.
2. Instantiate one cylinder.
3. Instantiate one cone.
4. Store the class instances into an array named "shapeArray."
5. Loop through the array and print out the instance data of each object using the object instance's "toString" method.

Show a Unified Modeling Language (UML) diagram of your project.

Condense your source files and UML diagram as a zipped folder and submit the zipped file as your critical thinking assignment (CTA).

## Option #2: Creating an Abstract Class Shape II

Create an abstract class "Shape" with the following specifications:

1. an abstract method "area()" of return type double
2. an abstract method "perimeter()" of return type double.

Put your code in a Java source file named "Shape.java."

### I.

Create a class "Rectangle" that extends the Shape class with the following specifications:

1. Attributes:  
width, length
2. Constructor:  
Implement a parameterized constructor needed initialize the data.
3. toString:  
Implement a "toString" method that prints out the area and perimeter.

Implement methods to compute the area and perimeter.

### II.

Create a class "Circle" that extends the Shape class with the following specifications:

1. Attributes:  
radius
2. Constructor:  
Implement a parameterized constructor to initialize the data.
3. toString:  
Implement a "toString" method that prints out the area and perimeter.

Implement methods to compute the area and perimeter.

### III.

Create a class "Triangle" that extends the Shape class with the following specifications:

1. Attributes:  
length of side1, length of side2, length of side3, each is a double
2. Constructor:  
Implement a parameterized constructor to initialize the data.
3. toString:  
Implement a "toString" method that prints out the area and perimeter.

Implement methods to compute the area and perimeter.

### IV.

Create a driver class named "ShapeArray" with the following specifications:

1. Instantiate one triangle.
2. Instantiate one circle.
3. Instantiate one square.

4. Store the class instances into an array named "shapeArray."
5. Loop through the array and print out the instance data of each object
6. using the object instance's "toString" method.

Show a UML diagram of your project.

Zip up your source files and UML diagram into a single zip file and submit the zip file as your CTA.

### **Mastery Exercise (10 points)**

### **Portfolio Milestone (25 points)**

Make appropriate corrections to the code you submitted in Modules 1-3 for your Critical Thinking Assignments. Corrections should reflect feedback from your instructor and improvements in execution, organization, and style. Resubmit your programs from Modules 1-3 with all outlined corrections by the end of this week.

## **Module 5**

### **Readings**

- Chapter 12 in *Programming in Java with ZyLabs*

### **Opening Exercise (0 points)**

### **Discussion (25 points)**

### **Critical Thinking (80 points)**

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

#### **Option #1: Implementing Recursion to Provide a Product**

Using recursion, create a program that will allow a user to enter five numbers. The program will provide the product of all five numbers using recursive methods.

Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

#### **Option #2: Implementing Recursion to Provide a Sum**

Using recursion, create a program that will allow a user to enter five numbers. The program will provide the sum of all five numbers using recursive methods.

Submit screenshots of your program's execution and output. Include all appropriate source code in a zip file.

### **Mastery Exercise (10 points)**

## **Module 6**

### **Readings**

- Chapter 15 in *Programming in Java with ZyLabs*
- Du Preez, B. (n.d.). *The simple big-o notation post*. Retrieved from <https://www.javacodegeeks.com/2011/04/simple-big-o-notation-post.html>
- Binary search. (n.d.). Retrieved from <https://www.geeksforgeeks.org/binary-search/>

- Linear search. (n.d.). Retrieved from <https://www.geeksforgeeks.org/linear-search/>

### **Opening Exercise (0 points)**

### **Discussion (25 points)**

### **Critical Thinking (50 points)**

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

#### **Option #1: Storing to an ArrayList I**

Create a Java program that will store 10 student objects in an ArrayList, ArrayList<Student>. A student object consists of the following fields:

- int rollno
- String name
- String address

Implement two comparator classes to sort student objects by name and by rollno (roll number).

Implement your own selection sort method and place your code in a separate Java source file. Do not use a sort method from the Java collections library.

#### **Option #2: Storing to an ArrayList II**

Create a Java program that will store 10 student objects in an ArrayList, ArrayList<Student>. A student object consists of the following fields:

- int rollno
- String name
- String address

Implement a comparator class to sort student objects by rollno (roll number). Implement your own merge sort method and place your code in a separate Java source file. Do not use a sort method from the Java collections library.

### **Mastery Exercise (10 points)**

## **Module 7**

### **Readings**

- Chapter 15 in *Programming in Java with ZyLabs*
- List Interface in Java with Examples. (n.d.). Retrieved from <https://www.geeksforgeeks.org/list-interface-java-examples/>
- Set in Java. (n.d.). Retrieved from <https://www.geeksforgeeks.org/set-in-java/>

### **Opening Exercise (0 points)**

### **Discussion (25 points)**

### **Mastery Exercise (10 points)**

### **Portfolio Milestone (25 points)**

Make appropriate corrections to the code you submitted in Modules 5-6 for your Critical Thinking Assignments. Corrections should reflect feedback from your instructor and improvements in execution,

organization, and style. Resubmit your programs from Modules 5-6 with all outlined corrections by the end of this week.

## **Module 8**

### **Readings**

- Chapter 13 in *Programming in Java with ZyLabs*
- Oracle. (2014). Queue implementations. *The Java Tutorials*. Retrieved from <https://docs.oracle.com/javase/tutorial/collections/implementations/queue.html>
- Exceptions in Java. (n.d.). Retrieved from <https://www.geeksforgeeks.org/exceptions-in-java/>
- Types of Exception in Java with Examples. (n.d.). Retrieved from <https://www.geeksforgeeks.org/types-of-exception-in-java-with-examples/>

### **Opening Exercise (0 points)**

### **Discussion (25 points)**

### **Mastery Exercise (10 points)**

### **Portfolio Project (300 points)**

Choose one of the following two assignments to complete this week. Do not do both assignments. Identify your assignment choice in the title of your submission.

#### **Option #1: Portfolio Project I: Lessons Learned and Final Program**

##### **Milestones**

Milestone 1 (due at the end of Week 4): Java source code (with corrections as required) for programs created in Modules 1-3.

Milestone 2 (due at the end of Week 7): Java source code (with corrections as required) for programs created in Modules 5-6.

##### **Final Portfolio Project**

In Week 8, the components you must complete for your Portfolio Project are the Lessons Learned Reflection and the Final Program. Carefully review the requirements below.

##### **Lessons Learned Reflection**

Write a summary outlining the lessons you have learned in this programming course. Reflect on how these lessons can be applied to effective coding.

This essay portion of your Portfolio must:

- Be 2-3 pages in length (not including the required title and references pages)
- Be formatted according to the CSU-Global Guide to Writing and APA Requirements
- Include at least three outside references from the course readings, formatted according to APA requirements.

##### **Final Program**

Write a Java program that incorporates a loop that prompts the user for student data. Student data are private fields in a student class including:

- String name
- String address
- double GPA

Each student object is stored in a linked list.

After the user completes the data entry, output the contents of the linked list in ascending sorted order by name to a regular text file that can be opened and viewed using a simple plain-text editor such as notepad.

Validate numeric data for Grade Point Average (GPA).

Compile your Lessons Learned Reflection, source code, screenshots of the application executing, and results into a single document.

Format your document in MS Word, according to CSU-Global Guide to Writing and APA Requirements, particularly in developing your Lessons Learned Reflection. Support your reflection with a minimum of three references, as noted above. Include both a cover page and a reference page with your Portfolio Project.

Submit your completed Portfolio Project by the end of Week 8.

## **Option #2: Portfolio Project II: Lessons Learned and Final Program**

### **Milestones**

Milestone 1 (due at the end of Week 4): Java source code (with corrections as required) for programs created in Modules 1-3.

Milestone 2 (due at the end of Week 7): Java source code (with corrections as required) for programs created in Modules 5-6.

### **Final Portfolio Project**

In Week 8, the components you must complete for your Portfolio Project are the Lessons Learned Reflection and the Final Program. Carefully review the requirements below.

#### **Lessons Learned Reflection**

Write a summary outlining the lessons you have learned in this programming course. Reflect on how these lessons can be applied to effective coding.

This essay portion of your Portfolio must:

- Be 2-3 pages in length (not including the required title and references pages)
- Be formatted according to the CSU-Global Guide to Writing and APA Requirements
- Include at least three outside references from the course readings, formatted according to APA requirements.

#### **Final Program**

Write a Java program that incorporates a loop that prompts a user for vehicle data. Vehicle data are private fields in a Vehicle class that include:

- String make
- String model
- double miles-pre-gallon

Each vehicle object is stored in a linked list.

After the user completes the data entry, output the contents of the linked list in ascending sorted order by miles-per-gallon to a text file that can be opened and viewed using a simple plain-text editor such as notepad.

Validate numeric data for miles-per-gallon.

Compile your Lessons Learned Reflection, your source code, screenshots of the application executing and results into a single document.

Format your document according to CSU-Global Guide to Writing and APA Requirements, particularly in developing your Lessons Learned Reflection, and support your reflection with a minimum of three references, as noted above. Include both a cover page and a reference page with your Portfolio Project.

Submit your completed Portfolio Project by the end of Week 8.

---

## COURSE POLICIES

---

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

### Course Grading

20% Discussion Participation  
0% Opening Exercises  
8% Mastery Exercises  
37% Critical Thinking Assignments  
35% Final Portfolio Project

---

## IN-CLASSROOM POLICIES

---

For information on late work and incomplete grade policies, please refer to our [In-Classroom Student Policies and Guidelines](#) or the Academic Catalog for comprehensive documentation of CSU-Global institutional policies.

### **Academic Integrity**

Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /repurposing your own work (see CSU-Global Guide to Writing & APA for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU-Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the Library and internet resources.

### **Citing Sources with APA Style**

All students are expected to follow the CSU-Global Guide to Writing & APA when citing in APA (based on the most recent APA style manual) for all assignments. A link to this guide should also be provided within most assignment descriptions in your course.

### **Disability Services Statement**

CSU-Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email [ada@CSUGlobal.edu](mailto:ada@CSUGlobal.edu) for additional information to coordinate reasonable accommodations for students with documented disabilities.

### **Netiquette**

Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults, or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments of an ethnic, religious, age, disability, sexist (or sexual orientation), or racist nature; and the unwanted sexual advances or intimidations by email, or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, please let your instructor know.