

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 places a heavy emphasis on students' ability to develop secure and functional computer programs using either Java or C++ programming languages. Students will use programming knowledge to complete programming projects based on real-world scenarios that reflect problems in most organizations. Additionally, students will check the security posture of the code by performing checks during development that will be documented and mitigated. Students will be covering topics and concepts such as ensuring security and functionality of computer programs.

COURSE OVERVIEW:

This course will focus on understanding various concepts that can be utilized in order to create software applications that are more secure. Students will learn how to identify common programming errors that can lead to software vulnerabilities for an organization. Students will be able to identify the risks and consequences associated with poor programming constructs. They will also apply the principles of strong software design and development to the Java and C++ programming languages.

COURSE LEARNING OUTCOMES:

1. Develop computer programs using C++ programming concepts.
2. Develop computer programs using Java programming concepts.
3. Demonstrate how to analyze security vulnerabilities in C++ and Java.
4. Write software programs that manage resources securely in different operating system environments.
5. Develop computer programs effectively using different programming languages and programming constructs to solve business problems.

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:

- Ballman, A. (2016). *SEI CERT C++ coding standard: Rules for developing safe, reliable, and secure systems in C++*. Software Engineering Institute (Carnegie Mellon University). Hanscom, MA.
- Eck, D. J. (2015). *Introduction to programming using Java* (7th ed.). CC BY-NC-SA 3.0. Retrieved from <http://math.hws.edu/javanotes/>
- Long, F., Mohindra, D., Seacord, R. C., Sutherland, D. F., & Svoboda, D. (2014). *Java coding guidelines: 75 recommendations for reliable and secure programs*. Upper Saddle River, NJ: Pearson Education. ISBN-13: 9780321933157
- Eclipse IDE for C/C++ Developers - <https://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/neonr>

Suggested:

- Backman, K. (2012). *Structured programming with C++*. Bookboon.com. Retrieved from <http://bookboon.com/en/structured-programming-with-c-plus-plus-ebook>

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 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

- Chapter 1 in *SEI CERT C++ Coding Standard: Rules for Developing Safe, Reliable, and Secure Systems in C++*
- Elsabagh, M., Fleck, D. & Stavrou, A. (2017). Strict Virtual Call Integrity Checking for C++ Binaries. *In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '17)*. ACM, New York, NY, USA, 140-154.
- Eck, D. J. (2015). *Introduction to programming using Java* (7th ed., version 7.0.1). CC BY-NC-SA 3.0.

Opening Exercise (0 points)

Discussion (25 points)

In this module, we will be introduced to working with the C++ programming language. What are some of the issues that you had installing and running the Eclipse IDE for C/C++ developers? Working with the C++ language, discuss the different data types available. What are some of the differences, if any, between data types in C++ versus Java? What tips can be utilized to identify possible vulnerabilities using C++ data types? Be sure to provide an appropriate source code example to illustrate your points.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Critical Thinking (60 points)

Option 1: Console Application and Syntax Corrections (60 points)

Demonstrate an understanding of basic C++ programming concepts by completing the following:

1. Create a simple C++ console application using Eclipse IDE that will accomplish the following:
 - a. Prints the following information for a fictional person:
 - i. First Name,
 - ii. Last Name,
 - iii. Street Address
 - iv. City
 - v. Zip code
2. Given the provided code in file `CSC450_CT1_mod1-1.cpp`, correct all syntax errors so that the code will compile correctly.
3. Given the provided code in file `CSC450_CT1_mod1-2.cpp`, correct all syntax errors so that the code will compile correctly.

Compile and submit source code(s) and screenshots of the application executing the results in a single zip file.

Option 2: Console Application and Syntax Corrections (60 points)

Demonstrate an understanding of basic C++ programming concepts by completing the following:

1. Create a simple C++ console application using Eclipse IDE that will accomplish the following:
 - a. Prints the following information for a fictional person:
 - i. Father Name,
 - ii. Mother Name,
 - iii. Country of Citizenship
 - iv. City of resident
 - v. Age
2. Given the provided code in file CSC450_CT1_mod1-1.cpp, correct all syntax errors so that the code will compile correctly.
3. Given the provided code in file CSC450_CT1_mod1-2.cpp, correct all syntax errors so that the code will compile correctly.

Compile and submit source code(s) and screenshots of the application executing the results in a single zip file.

Mastery Exercise (10 points)

Portfolio Milestone Reminder

Option 1: For your Portfolio Project in this course, which is in two parts, you will be programming code to demonstrate concurrency concepts using both C++ and in Java. You will complete the C++ code in Module 7 as a milestone to the project; then you will develop Java code and a written comparison and analysis of the performance implementations of both programming languages in Module 8. Review the complete prompts and rubrics for these Portfolio Project assignments in the Module 7 and Module 8 materials folders.

Option 2: For your Portfolio Project in this course, which is in two parts, you will be programming code to demonstrate concurrency concepts using both C++ and in Java. You will complete the C++ code in Module 7 as a milestone to the project; then you will develop Java code and a written comparison and analysis of the performance implementations of both programming languages in Module 8. Review the complete prompts and rubrics for these Portfolio Project assignments in the Module 7 and Module 8 materials folders.

MODULE 2

Readings and Videos

- Chapter 6 in *SEI CERT C++ Coding Standard: Rules for Developing Safe, Reliable, and Secure Systems in C++*
- Backman, K. (2012). Chapter 5: Strings. In *Structured programming with C++* (pp. 96-120). Bookboon.com.
- Schmid, M. (2017). Finding Consensus Strings with Small Length Difference between Input and Solution Strings. *ACM Transactions on Computation Theory (TOCT)*, 9(3), 1-18. doi: <https://doi-org.csuglobal.idm.oclc.org/10.1145/3110290>

Opening Exercise (0 points)

Discussion (25 points)

Working with strings can lead to various security flaws and errors in software development using the C++ language. What are the common string manipulation errors that can be encountered? How can these errors be resolved and/or limited? What tips can be utilized to identify security vulnerabilities related to strings in C++? Be sure to provide an appropriate source code example to illustrate your points.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Critical Thinking (60 points)

Option 1: String Input Console Application and Program Analysis (60 points)

Demonstrate an understanding of basic C++ programming concepts by completing the following:

1. Program: Create a simple C++ console application that will write a program that will take string input from a user. Your program should reverse the string and then print the output to the screen. Take input from the user 3 times for varying string lengths.
2. Program Analysis: Given your program implementation, discuss and identify the possible vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Then discuss and identify possible problems that can result in errors for string manipulation of data. Your program analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

Option 2: String Input Console Application and Program Analysis (60 points)

Demonstrate an understanding of basic C++ programming concepts by completing the following:

1. Program: Create a simple C++ console application that will write a program that will take two string inputs from a user. Your program should concatenate the two strings and then print the resulting output to the screen. Take the two string inputs from the user 3 times for varying string lengths.
2. Program Analysis: Given your program implementation, discuss and identify the possible vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Then discuss and identify possible problems that can result in errors for string manipulation of data. Your program analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

Mastery Exercise (10 points)

Readings

- Chapter 9 in *SEI CERT C++ Coding Standard: Rules for Developing Safe, Reliable, and Secure Systems in C++*
- Chapter 3 & 4 in *Secure Coding in C and C++*
- Maalej, M., Paisante, V., Pereira, F.M.Q., & Gonnord, L. (2018). Combining range and inequality information for pointer disambiguation. *Science of Computer Programming*, 152(2018), 161-184.
- Princeton (n.d.). *Pointers and arrays: CS 217*. [PowerPoint].

Opening Exercise (0 points)

Discussion (25 points)

In this module, we will explore working with pointers in the C++ programming language. How are pointers used with arrays and what are the benefits to utilizing pointers? Are arrays necessary when using a pointer? Additionally, what is the difference between the *address-of* operator and the *dereference* operator in C++? Be sure to provide an appropriate example to illustrate your viewpoints.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Mastery Exercise (10 points)

Critical Thinking (60 points)

Option 1: Integer Pointers Program and Analysis (60 points)

Demonstrate an understanding of basic C++ programming concepts by completing the following:

1. Program: Create a C++ program that asks the user to enter three integer values as input. Store the values into three different variables. For each variable, create three integer pointers that point to each value. Display the contents of the variables and pointers. In your program, be sure to use the new operator and delete operators to management memory.
2. Program Analysis: Given your program implementation, discuss and identify in possible security vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Discuss and identify possible problems that can result in errors when using integer pointers. Your analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

Option 2: Integer Pointers Arithmetic Program and Analysis (60 points)

Demonstrate an understanding of basic C++ programming concepts by completing the following:

1. Program: Create a C++ program that asks the user to enter three integer values as input. Store the values into three different variables. For each variable, create three integer pointers that point to each value. Perform two different pointers arithmetic and display the contents of the

variables and pointers. If the arithmetic is not defined, display an error message. In your program, be sure to use the new operator and delete operators to management memory.

2. Program Analysis: Given your program implementation, discuss and identify in possible security vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Discuss and identify possible problems that can result in errors when using integer pointers. Your analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

MODULE 4

Readings

- Chapter 9 in *SEI CERT C++ Coding Standard: Rules for Developing Safe, Reliable, and Secure Systems in C++*
- Chapter 5 & 6 in *Secure Coding in C and C++*
- Demidov, R., Pechenkin, A., and Zegzhda. P. (2017). Integer overflow vulnerabilities detection in software binary code. *In Proceedings of the 10th International Conference on Security of Information and Networks (SIN '17)*. ACM, New York, NY, USA, 101-106.
- Coker, Z., Hasan, S, Overbey, J., Hafiz, M. & Kastner, C. (n.d.) *Integers in C: An open invitation to security attacks?*
- Storming Robots_(n.d.) *Basic in output formatting with C++ stream method- cout/cin*

Opening Exercise (0 points)

Discussion (25 points)

In creating C++ applications, you can utilize various formatting functions in the iostream library. What are some of the formatting vulnerabilities that can be encountered in using the iostream library in C++? What tips can be utilized to identify these vulnerabilities? Be sure to provide an appropriate source code example to illustrate your points.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Mastery Exercise (10 points)

Critical Thinking (70 points)

Option 1: Employee Salary Calculator Program and Analysis (70 points)

Demonstrate an understanding of C++ programming concepts by completing the following:

1. Program: Create a C++ program that will function as an Employee Salary Calculator. Obtain from the user:
 - a. Double variables for
 - i. Standard Hours worked
 - ii. Rate of Pay
 - iii. Overtime Hours (if applicable)
 - b. Calculate appropriate pay for standard hours and overtime (1.5 x rate of pay).

- c. Provide formatted output using the following methods
 - i. `setw()`
 - ii. `setprecision()`
 2. Store the values into three different variables. For each variable, create three integer pointers that point to each values. Display the contents of the variables and pointers. In your program, be sure to use the new operator and delete operators to manage dynamic memory.
 3. Program Analysis: Given your program implementation, discuss and identify in possible security vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Discuss and identify possible problems that can result in errors when using formatted output. Your analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

Option 2: Employee Commission Calculator Program and Analysis (70 points)

Demonstrate an understanding of C++ programming concepts by completing the following:

1. Program: Create a C++ program that will function as an Employee Commission Calculator. Obtain from the user:
 - a. Double variables for
 - i. Commission
 - ii. Base salary
 - iii. Overtime Hours (if applicable)
 - b. Calculate appropriate pay by adding the commission to the base salary (make sure you multiply the commission by base salary first to get the dollar amount).
 - i. `setw()`
 - ii. `setprecision()`
1. Store the values into three different variables. For each variable, create three integer pointers that point to each values. Display the contents of the variables and pointers. In your program, be sure to use the new operator and delete operators to manage dynamic memory.
2. Program Analysis: Given your program implementation, discuss and identify in possible security vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Discuss and identify possible problems that can result in errors when using formatted output. Your analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

MODULE 5

Readings

- Chapter 8 & 11 in *SEI CERT C++ Coding Standard: Rules for Developing Safe, Reliable, and Secure Systems in C++*

- Chapters 7 & 8 in *Secure Coding in C and C++*
- Tripp, C., Hyde, D., & Grossman-Ponemon, B. (2018). FRC: A high-performance concurrent parallel deferred reference counter for C++. *In Proceedings of the 2018 ACM SIGPLAN International Symposium on Memory Management (ISMM 2018)*. 14-28. doi: <https://doi-org.csuglobal.idm.oclc.org/10.1145/3210563.3210569>
- Gu, R., Gan, B., Zhao, J., Ning, Y., Cui, H. & Yang, J. (n.d.) *Understanding and detecting concurrency attacks*

Opening Exercise (0 points)

Discussion (25 points)

Concurrency in C++ applications can lead to a number of different vulnerabilities based upon the incorrect implementation of application threads. What are the best choices to make to avoid common concurrency vulnerabilities and pitfalls? Be sure to provide an appropriate source code example to illustrate your points.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Mastery Exercise (10 points)

Critical Thinking (60 points)

Option 1: User Input Program and Analysis (60 points)

Demonstrate an understanding of C++ programming concepts by completing the following:

1. Program: Create a C++ program that will obtain input from a user and store it into the provided CSC450_CT5_mod5.txt file. Your program should append it to the provided text file, without deleting the existing data:
 - a. Store the provided data in a CSC450_mod5-2.txt file.
 - b. Create a reversal method that will reverse all of the characters in the CC450-mod5_2.txt file and store the result in a CSC450-mod5-reverse.txt file.
2. Program Analysis: Given your program implementation, discuss and identify in possible security vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Discuss and identify possible problems that can result in errors for string manipulation of data. Your analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.

Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

Option 2: User Input/Search Program and Analysis (60 points)

Demonstrate an understanding of C++ programming concepts by completing the following:

1. Program: Create a C++ program that will obtain input from a user and store it into the provided CSC450_CT5_mod5.txt file. Your program should append it to the provided text file, without deleting the existing data:

- a. Store the provided data in a CSC450_mod5-2.txt file.
- b. Search the file for your name returning it if found or returning the message "name not found" in CSC450-not-found.txt file

2. Program Analysis: Given your program implementation, discuss and identify in possible security vulnerabilities that may exist. If present, discuss solutions to minimize the vulnerabilities. Discuss and identify possible problems that can result in errors for string manipulation of data. Your analysis should be 1-2 pages in length.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

MODULE 6

Readings

- Chapter 1 in *Java Coding Guidelines*
- Meng, N., Nagy, S., Yao, D., Zhuang, W., & Argoty, G.A. (n.d.). *Secure coding practices in Java: Challenges and vulnerabilities*

Opening Exercise (0 points)

Discussion (25 points)

Data sensitivity can have a detrimental effect on the performance of a Java application. What are some factors that affect the sensitivity of data in a Java application? What might be an example of an application that contains sensitive data in a noncompliant manner? Be sure to provide an appropriate source code example to illustrate your points.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Mastery Exercise (10 points)

Critical Thinking (60 points)

Option 1: Compliant/Noncompliant Solutions (60 points)

The following code segment was provided in Chapter 1 of Java Coding Guidelines:

```
void readData() throws IOException{
    BufferedReader br = new BufferedReader(new InputStreamReader(
        new FileInputStream("file")));
    // Read from the file
    String data = br.readLine();
}
```

The code is presented as a noncompliant code example. A compliant solution is presented for this example. For this assignment, identify two additional compliant solutions that can be utilized to ensure

the protection of sensitive data. Provide a detailed explanation of factors that influence noncompliant code and specifically how your solutions are now compliant. Your explanation should be 2-3 pages in length.

Submit the following components:

- Word document with appropriate program analysis for your compliant solutions and source code in the word file.
- Submit your .java source code file(s). If more than 1 file, submit a zip file.

Option 2: Compliant/Noncompliant Solutions (60 points)

The following code segment was provided in Chapter 1 of Java Coding Guidelines:

```
void readData() throws IOException{  
    BufferedReader br = new BufferedReader(new InputStreamReader(  
        new FileInputStream("file")));  
    // Read from the file  
    String data = br.readLine();  
}
```

The code is presented as a noncompliant code example. A compliant solution is presented for this example. For this assignment, identify two additional noncompliant solutions to demonstrate further the danger of not protecting sensitive data. Provide a detailed explanation of factors that influence compliant code and specifically how your solutions contradict them. Your explanation should be 2-3 pages in length.

Submit the following components:

- Word document with appropriate program analysis for your compliant solutions and source code in the word file.
- Submit your .java source code file(s). If more than 1 file, submit a zip file.

MODULE 7

Readings

- Chapter 2 in *Java Coding Guidelines*
- Sun Microsystems, Inc. (2005). *Java security overview*. [White paper].

Opening Exercise (0 points)

Discussion (25 points)

What are the benefits of scope minimization in Java? Why is it a good technique to practice? Provide an appropriate source code example to illustrate your points.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Mastery Exercise (10 points)

Portfolio Milestone (75 points)

Option 1: PORTFOLIO PROJECT MILESTONE (75 points)

For your Portfolio Project in this course, which is in two parts, you will be programming code to demonstrate concurrency concepts using both C++ and in Java. You will complete the C++ code in Module 7 as a milestone to the project; then you will develop Java code and a written comparison and analysis of the performance implementations of both programming languages in Module 8. Review the complete prompts and rubrics for these Portfolio Project assignments in Module 7 and Module 8 materials folders.

Portfolio Project: Part I

For your Portfolio Project, you will demonstrate an understanding of the various concepts discussed in each module. For the first part of your Portfolio Project, you will create a C++ application that will exhibit concurrency concepts. Your application should create two threads that will act as counters. One thread should count up to 20. Once thread one reaches 20, then a second thread should be used to count down to 0. For your created code, provide a detailed analysis of appropriate concepts that could impact your application. Specifically, address:

- Performance issues with concurrency
- Vulnerabilities exhibited with use of strings
- Security of the data types exhibited.

For this milestone to your Portfolio Project, submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

Option 2: PORTFOLIO PROJECT MILESTONE (75 points)

For your Portfolio Project in this course, which is in two parts, you will be programming code to demonstrate concurrency concepts using both C++ and in Java. You will complete the C++ code in Module 7 as a milestone to the project; then you will develop Java code and a written comparison and analysis of the performance implementations of both programming languages in Module 8. Review the complete prompts and rubrics for these Portfolio Project assignments in the Module 7 and Module 8 materials folders.

Portfolio Project: Part I

For your Portfolio Project, you will demonstrate an understanding of the various concepts discussed in each module. For the first part of your Portfolio Project, you will create a C++ application that will exhibit concurrency concepts. Your application should create two threads that will act as counters. One thread should count down to 0. Once thread one reaches 0, then a second thread should be used to count up to 100. For your created code, provide a detailed analysis of appropriate concepts that could impact your application. Specifically, address:

- Performance issues with concurrency
- Vulnerabilities exhibited with use of strings
- Security of the data types exhibited.

For this milestone to your Portfolio Project, submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .cpp source code file(s). If more than 1 file, submit a zip file.

MODULE 8

Readings

- Chapter 3 in *Java Coding Guideline*
- Goffi, A., Gorla, A., Mattavelli, A., & Pezzè, M. (n.d.). *Intrinsic redundancy for reliability and beyond*

Opening Exercise (0 points)

Discussion (25 points)

Handling exceptional conditions can lead to code that can lead to control flow issues and performance degradation within a Java application. In regard to developing appropriate exceptions in Java, what are some best practices to follow? How can it be determined when it is optimal to use a user-defined exception with an appropriate try/catch statement? What security and performance issues should be analyzed? Be sure to provide an appropriate source code example to illustrate your points.

Be sure to post an initial, substantive response by Thursday at 11:59 p.m. MST and respond to 2 or more peers with substantive responses by Sunday at 11:59 p.m. MST. A substantive initial post answers the question presented completely and/or asks a thoughtful question pertaining to the topic. Substantive peer responses ask a thoughtful question pertaining to the topic and/or answer a question (in detail) posted by another student or the instructor.

Mastery Exercise (10 points)

Portfolio Project (275 points)

Option 1: Portfolio Project: Part II (275 points)

For your Portfolio Project, you will demonstrate an understanding of the various concepts discussed in each module. For the second part of your Portfolio Project, you will create a Java application that will exhibit concurrency concepts. Your application should create two threads that will act as counters. One thread should count up to 20. Once thread one reaches 20, then a second thread should be used to count down to 0. For your created code, please provide a detailed analysis of appropriate concepts that could impact your application. Specifically, please address:

- Performance issues with concurrency
- Vulnerabilities exhibited with use of strings
- Security of the data types exhibited.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the word file.
- Submit your .java source code file(s). If more than 1 file, submit a zip file.
- Provide a detailed comparison between the performance implementations between the Java and C++ versions of your applications. Which implementation may be considered less vulnerable to security threats and why? Your detailed comparison should be 3-4 pages in length and should be formatted according to the CSU-Global Guide to Writing and APA.

Option 2: Portfolio Project: Part II (275 points)

For your Portfolio Project, you will demonstrate an understanding of the various concepts discussed in each module. For the second part of your Portfolio Project, you will create a Java application that will exhibit concurrency concepts. Your application should create two threads that will act as counters. One thread should count down to 0. Once thread one reaches 0, then a second thread should be used to count up to 100. For your created code, please provide a detailed analysis of appropriate concepts that could impact your application. Specifically, please address:

- Performance issues with concurrency
- Vulnerabilities exhibited with use of strings
- Security of the data types exhibited.

Submit the following components:

- Word document with appropriate screenshots of your program executing, program analysis responses, and source code in the Word file.
- Submit your .java source code file(s). If more than 1 file, submit a zip file.
- Provide a detailed comparison between the performance implementations between the Java and C++ versions of your applications. Which implementation may be considered less vulnerable to security threats and why? Your detailed comparison should be 3-4 pages in length and should be formatted according to the CSU-Global Guide to Writing and APA.

COURSE POLICIES

Course Grading

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

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

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 and APA Requirements* 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 and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on CSU-Global APA style, please review the APA resources within the CSU-Global Library under the “APA Guide & Resources” link. A link to this document 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 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.