

Introduction to Programming I

Course of Study for AAT1

Description

This course of study outlines the sequence of learning activities to help you develop competence in the subject area of Introduction to Programming I. Your competence will be assessed as you complete a series of performance tasks for performance assessment AAT1. This course of study may take up to 11 weeks to complete depending on your educational background, work experience, and hours that you are able to dedicate to your studies. Consult with your mentor if you wish to accelerate your progress through this course of study. It is important that you follow the activities sequentially as you prepare for your assessment. This tool is also designed to help you become an independent learner by providing multiple learning methods.

Introduction

Welcome to Introduction to Programming Part I. My name is Cindy Delia, and I will serve as your mentor as you work through this course of study. I have over 20 years of experience in IT. My background in IT includes experience as a systems analyst and designer, database administrator, and a corporate trainer. I am a certified instructor and have over 10 years of experience teaching computer courses. My academic background includes a master's degree in information technology and a bachelor's degree in business and information technology. I am a doctoral candidate in business and computer science at NorthCentral University. I am eager to assist you with this course of study and will contact you periodically to answer questions and check your progress.

Course of Study Mentor.....Cindy Delia
Emailcdelia@wgu.edu
Telephone..... 1-866-895-9660, x1850
Office HoursMonday, 9am - 5pm, Eastern Time
 Tuesday, 9am - 5pm, Eastern Time
 Wednesday, 9am - 5pm, Eastern Time
 Thursday, 9am - 5pm, Eastern Time
 Friday, 9am - 5pm, Eastern Time

This document outlines the sequence of steps (reading, writing code, completing activities, and, taking assessments) required to develop the necessary competence in introductory programming using Java.

Competence is measured by the successful completion of the AAT1 performance tasks located in TaskStream and the course of study tasks. It is strongly recommended that you complete the activities to ensure your mastery of the required knowledge and competence to satisfactorily pass this course of study.

The activities are designed and written so that their completion produces a portfolio of software programs you can present to potential employers. This will demonstrate that you know how to create software applications that meet requirements.

Java applications can run on any platform which is why it is well suited for applications both big and small. Java is free, and it is everywhere. All of these reasons are why Java software developers are in high demand and can command an excellent salary. Being fluent in the Java programming language, knowledgeable of the

Java API classes, and skilled in software development will qualify you to design and develop applications in a wide range of industries including education, energy, ecommerce, telecommunications, gaming, health care, and finance.

There are several academic competencies associated with this course of study. These will be addressed in the coming weeks.

Competency 430.1.1: Control Structures

The graduate develops working programs that use appropriate control structures and accurately evaluates paths in program code.

Competency 430.1.2: Algorithm Design and Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

Learning Resources:

- Hortsman, C. (2008). *Big Java* (3rd ed.). Wiley. ISBN: 9780470105542.
- NetBeans Integrated Development Environment (for editing and debugging your code)
- Java SE Developer's Kit (The Java language and environment can be bundled with NetBeans.)

Week 1

Preparing for Success

To successfully complete AAT1, you will need the appropriate learning resources. You should also prepare a calendar to schedule time devoted to your studies. Share your calendar with family and friends so they are aware of your obligations.

Acquire Learning Resources

Arrange to obtain the learning resources listed below so there will be no delays in your studies. These items are essential for you and this document will guide you week-by-week in the use of these materials.

Obtain *Big Java* Textbook

Access the learning resource *Big Java*, 3rd edition, by Cay Horstmann, ISBN: 9780470105542.

This is the required learning resource for Introduction to Programming Part I and II. It is not necessary for you to purchase this text as you can access this learning resource online through your SkillSoft Books24x7 account. If you wish to purchase a hard copy of this textbook, you may also choose to buy a hard copy from a bookstore or online at your own cost. Please use the ISBN number(s) to make sure that you order the correct edition. You can also purchase the textbook via the WGU Bookstore (EdMap).

Install NetBeans Onto Your Computer

URL: <http://www.netbeans.org>

You will be required to develop Java applications for Introduction to Programming Part I and Part II. NetBeans is the recommended Integrated Development Environment (IDE). This IDE will provide you an editor, debugger, and the Java Developers Kit. If you use NetBeans, you will not have to download the Java SE Developer's Kit as a separate activity. You can download the SDK bundled with NetBeans (recommended).

Follow the link above to download, and then follow the installation instructions for Java SE NetBeans IDE Download Bundle.

□ Create Notebook

It is suggested that you create a paper or digital notebook as you work through the AAT1 Course of Study. Use organizers or dividers to separate your work. Suggested dividers include:

- Java syntax notes
- Solutions to compiler errors encountered
- NetBeans commands
- Java API classes used

Getting Familiar With the Development Environment

You will learn how to create a new Java project in NetBeans and also get familiar with using the debugger. The NetBeans website has some excellent tutorials to help you get started.

□ Your First Java Application

URL: <http://www.netbeans.org/kb/docs/java/quickstart.html>

In this activity you will create your first Java application. You will start by completing this fifteen-minute tutorial which will give you a quick but thorough introduction to developing applications using NetBeans. You will learn how to build a new project, use the editor, use the compiler, and then run the application.

Save your resulting HelloWorldApp project. You will use this during the next activity.

- What challenges did you run into while completing this tutorial?
- Did you run into any compiler errors? If you did, what were they?
- Try removing a semicolon from one of the end of the lines of your code. What happens when you try to build and run your application now?

□ Getting Familiar With the Debugger

`System.out.printf()` can be used to print formatted output to the screen. Int variables can be incremented by the `++` operator. This information will be helpful to your understanding of the modifications you will need to make to your HelloWorldApp project.

1. Open your HelloWorldApp project.
2. Add the following lines of code after the line of code that prints "Hello World!"

```
int i = 1;
System.out.printf("Hello %d\n", i++);
System.out.printf("Hello %d\n", i++);
System.out.printf("Hello %d\n", i++);
```

3. Add a breakpoint at the line that prints “Hello World!” Breakpoints can be turned on by toggling the line number.
 4. Open the watch local variables window. Window > Debugging > Local Variables.
 5. Run the application. Run > Debug Main Project.
 6. Once your program pauses at your breakpoint, notice the value of l in the “Watch Local Variables” window.
 7. Step to the next line.
 8. Run > Step Over.
 9. Continue to step over each statement until program ends.
- What do you think the following commands do?
 - Run > Step Over
 - Run > Step Into
 - Run > Continue
- What are two ways to set a break point?
 - What three ways can you initiate the Step Over and Step Into commands?
 - What does the Watch Local Variables window show you?
 - How can you use these debugger features to debug your applications?

Week 2

Basic Application Development

This lesson will help you understand the fundamentals of Java programming, basic syntax for Java, and learn about the Java Virtual Machine. Knowing basic syntax is important to developing applications that will compile and run.

Competency 430.1.2: Algorithm Design and Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

Developing Java Applications

This topic focuses on developing your understanding of Java syntax. Syntax must be exact in order for your applications to compile and run. As you complete the reading and write the suggested applications, pay close attention to the syntax for each statement. This topic will introduce you to the Java Virtual Machine which makes Java applications portable. This portability characteristic is one of the reasons why Java is so widely used. You will learn how to read in command line arguments and process them in an application. As you complete the readings and tasks for this topic, ask yourself these questions:

- Do you think Java is interpreted or compiled?
- How are command line arguments passed into a Java application?

□ Chapter 1: “Introduction”

Read chapter 1 in your textbook. Notice the syntax of all examples provided. Java syntax must be exact or your code will not compile.

□ Write Java Application with Command Line Arguments

Write a program that does the following:

1. Prints “Hello Name” where “Name” is a first name entered as a command line argument
- Did you run into any problems reading in the command line arguments?
 - Did you run into any problems printing out the name read in?

□ Write an Interactive Java Application

Write an interactive program that does the following:

1. Prompts you to enter your first name
 2. Prints “Hello Name” where “Name” is the first name entered at the prompt
- Did you run into any compiler errors? How did you resolve the errors?

Introduction to Objects and Classes

This lesson will provide you an overview of object-oriented design and development. You must have a solid understanding of objects and classes in order to develop applications that are maintainable and extensible.

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

Objects and Classes

This topic provides an overview of objects, methods, getters, setters, attributes, and the Java API Documentation. Encapsulation is a key strategy for developing applications that are maintainable and is achieved by providing getters and setters. Java provides a large library of reusable classes through the Java API. You will learn the location of the documentation of this important resource. As you complete the reading, note the differences between a class and an object.

□ Chapter 2: “Using Objects”

Read chapter 2 in your textbook. Note the link to the Sun Java API Documentation. See if you can complete review exercises R2.1–R2.13. Answer the following:

- What is a getter? What is a setter?
- What is an accessor method? What is a mutator method?

□ Bookmark Sun Java API Documentation

Locate the current version of the Sun Java API Documentation. You should consult the Sun Java API Documentation each time you use one of the classes from the API. Take a look around at the classes. Notice the information you can obtain for each class. Bookmark this link and refer to it often.

- Did you find the link to the Java API Documentation?
- What is the current version of the Java API?
- How do you look up a class or interface in the Java API Documentation?
- How do you determine which package to include for a class you will use?
- What kind of information is provided in the Java API Documentation?

Constructors

This topic provides an overview of constructing, implementing, and using classes which are the foundation of object-oriented applications. Learning how to develop constructors that appropriately set the state of an object is part of proper application development.

□ Chapter 3: “Implementing Classes”

Read chapter 3 in your textbook. Pay close attention to the concept of public interfaces, implementing constructors, and methods. Complete review exercises R3.1–R3.12.

- How do you create a constructor?
- What is a default constructor?
- When would you use the “this” keyword?

Week 3

Introduction to Algorithm Development

In this lesson you will learn how to use assignment, arithmetic, relational, and logical operators. Operators are key to algorithm development in your Java applications.

Competency 430.1.2: Algorithm Design and Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

Using Operators

This topic focuses on algorithm development for problem solving and using assignment, arithmetic, relational, and logical operators. Relational and logical operators are key to decision control structures which will be covered in an upcoming topic. Applications are often used for calculations which require the use of assignment and arithmetic operators.

□ Chapter 4: “Fundamental Data Types”

Read chapter 4 in your textbook. Learn each primitive data type and determine the size limits of each. Pay close attention to how operators can be used in algorithms. Complete review exercises R4.2, R4.9, R4.10, R4.11, R4.12, and R.14.

- What are the Java primitive types?
- Which of these primitives are numeric types?
- How would you increment and decrement ints? Are there shortcuts to do this?

Control Structures

This lesson will give you practice with using control structures to create execution paths. The majority of the applications you will develop will require the use of control structures to develop the logical flow of control of events, behavior, and actions.

Competency 430.1.1: Control Structures

The graduate develops working programs that use appropriate control structures and accurately evaluates paths in program code.

Decision Structures

The purpose of conditional statements is to execute a block of code based on whether a certain condition is true or false. If true, the block of code is executed; if false, the block of code is ignored. Decision structures are frequently required to develop program logic. As you complete the readings and tasks for this topic, ask yourself these questions:

- When should a conditional statement should be used?
- What types of conditional statements does Java provide?

□ Chapter 5: “Decisions”

Read chapter 5 in your textbook. See if you can type in some of the examples and get them to run. Complete review exercises: R5.1, R5.3, R5.5, R5.6, and R5.8.

- If you are unable to answer the review questions, create a simple program to try out the syntax or code segment in question.

□ Create an Application That Uses Switch and Break

Write a program that uses a break statement within a switch statement and does the following:

1. Declares variable of type int to hold value of month
2. Uses the Calendar class to get the value of the current month
3. Uses a switch statement to case on the value of month

Note: If the month is 1, 3, 5, 7, 8, 10, or 12, the program should print “This month has 31 days.” If the month is 4, 6, 9, or 11, the program should print “This month has 30 days.” If the month is 2, the program should print “This month has 28 days unless it is a leap year.”

4. Uses the fall-through technique on months with the same number of days to minimize the number of print statements needed
5. Uses the break statement to break out of the switch statement

Note: The switch statement should only have four break statements including the one used in the default clause.

6. Includes a default clause that handles an invalid month type

Week 4

Control Structures Continued

This lesson will give you practice with using control structures to create execution paths. The majority of the applications you will develop will require the use of control structures to develop the logical flow of control of events, behavior, and actions.

Competency 430.1.1: Control Structures

The graduate develops working programs that use appropriate control structures and accurately evaluates paths in program code.

Iteration

This topic demonstrates how to use for, while, and do loops. Computer applications are often developed to handle repetitive tasks. These control structures provide the ability to repeat program statements. As you complete the readings and tasks for this topic, ask yourself these questions:

- Do I know how to use a for loop?
- What is the difference between a while loop and a do-while loop?
- Do I have the syntax memorized each?

□ Chapter 6: “Iteration”

Read chapter 6 in your textbook. Be sure to follow each code example carefully. Complete review exercises R6.1 – R6.6.

- If you are unable to answer the review questions, create a simple program to try out the syntax or code segment in question.

□ Develop an Application That Uses a for Loop

For loops are one of the most frequently used control structures. Write a program that does the following:

1. Uses a for loop that steps through the numbers one through fifty
2. Uses the logical AND operator (&&) to check if the current number is odd and a multiple of five

Note: If the number meets both criteria, the number is printed. Each number should be printed on the same line with commas separating each, but there should be no comma after 45.

- Did you use single line and multiline comments in your code? Why is commenting your code a good idea? **From this point forward, include embedded comments in your code.**

□ Develop an Application That Uses a While Loop

While loops are another frequently used control structure. Write a program that does the following:

1. Declares and initializes a primitive data type of int to the value of five
2. Implements the following do-while loop:

```
int i = 5;
System.out.println("Do While");
do
{
    i++;
    System.out.println("i = " + i);
}
while (i<5);
System.out.println("i currently is = " + i);
```

3. Implements the following while loop. Reset primitive data type i back to the value of five:

```
i = 5;
while (i<5)
{
```



```
i++;  
System.out.println("i currently is = " + i);  
}
```

- What is the difference between a do-while and a while loop?

Week 5

Manipulating Strings

This lesson will provide you with an introduction to the string data and the many methods provided on the string class for manipulating strings.

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

Strings

Character strings are frequently encountered in data processed by Java applications. You must know how to search for characters in a string as well as know how to manipulate strings. There are several methods provided on the string class that let you manipulate strings. As you complete the readings and tasks for this topic, ask yourself these questions:

- Do I know how to reference the string class in the Java API Documentation?
- Can I find each string method used in the examples in the Java API Documentation?

□ Reference String Class in Java API Documentation

Reference the string class and note all constructors, attributes, and methods provided by the string class.

- Is it always necessary to call the constructor for a string? Why or why not?
- What package contains the string class?
- What do the following methods do?
- charAt()
- compareTo()
- contains()
- equals()
- indexOf()
- lastIndexOf()
- replace()
- toLowerCase()
- toUpperCase()
- trim()
- How would you determine the size of a string?

Week 6

Data Structures

This lesson will provide you with an introduction to data structures provided by the Java API and provide an understanding of how collections can be used. Data structures are a fundamental computer science topic that

has been studied for decades. You must know the characteristics of the different types of common data structures and how they can be useful in your applications. Most of your applications will be data intensive.

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

Collections

Collections allow you to store and manipulate a collection of objects. The Java API provides several reusable data structures that are efficiently written. Knowing how to use these collection types will save you time writing your own that will likely be less efficient. This topic introduces you to collection types Arrays and ArrayLists. As you complete the readings and tasks for this topic, ask yourself these questions:

- What are the different categories of collections provided by the Java API?
- What are the characteristics of each collection type?

□ Chapter 7: “Arrays and ArrayLists”

Read chapter 7 in your textbook. Note the differences between arrays and array Lists. Look up the array and array list classes in the Java API Documentation. Note the fields and methods for each. Complete review exercises R7.1–R7.4.

- What are the wrapper classes for byte, Boolean, char, double, float, int, long, and short?
- What does auto-boxing do? What version of Java was it introduced?
- How would you use a “for each” loop to iterate through the elements of an array?

□ Create an Application That Uses a Nested for Loop

Write a program that does the following:

1. Stores the numbers 1 through 50 in a two-dimensional array
2. Uses a nested for loop to print the numbers in the two-dimensional array with ten numbers per line

Note: The first row should display numbers one through ten.

Week 7 Classes

This lesson will provide you with an understanding of how to create well-designed classes that encapsulate data and behavior to produce easy to maintain applications. Classes are the foundation of object-oriented applications.

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

Proper Design of Classes

Classes should be created to encapsulate data and to create code that is cohesive and easy to maintain. When designing methods, you may want to use preconditions to restrict the parameters of a method or to ensure a method is invoked when the object is in the appropriate state. Proper class

design is key to capitalizing on principles that lead to well-designed systems. As you complete the reading, ask yourself the following questions:

- Do I know what is meant by loose coupling?
- Do I know what is meant by tightly coupled code?
- Do I understand how to use an assertion?
- What are the characteristics of each collection type?

□ Chapter 8: “Designing Classes”

Read chapter 8 in your textbook. Pay close attention to the concept of cohesion and coupling. Develop your understanding of scope and understand how it applies to local variables and class members. Pay close attention to the section on creating packages.

- How would you use post- and preconditions?
- How do you create a package?
- What are the benefits of using packages?
- What is a static method? What is a static field?

Week 8

More Complex Control Structures

Control structures are used to control the flow of a program when things go right and when things go wrong. In this section we are going to look at controlling flow when there is a problem at run time. This will allow you to write robust applications that can handle imperfect scenarios such as incorrect user entry or problems with the data.

Competency 430.1.1: Control Structures

The graduate develops working programs that use appropriate control structures and accurately evaluates paths in program code.

Exception Handling

Sometimes things go wrong at run time and exceptions are thrown. Customer satisfaction can be achieved by developing robust applications that gracefully handle issues at run time. In this topic, you will learn how to utilize exception handling to write robust applications that gracefully handle imperfect scenarios. As you complete the readings and tasks for this topic, ask yourself these questions:

- What are the different situations that can cause an exception?
- Why is exception handling important?

□ Chapter 11: “Input/Output and Exception Handling”

Read chapter 11, sections 11.2–11.6 in your textbook. As you read, note the syntax for try/catch blocks.

- What is the difference between a checked and an unchecked exception?
- What does finally do?
- How would you design your own exception type? Why would you want to do this?

More Complex Algorithm Design and Development

Oftentimes our algorithms require using data. We need to develop a way to read and write data from sources such as files, ports, databases, the screen, and the keyboard. Input and output using the Java API IO streams

is usually a challenging subject. In this lesson you will focus on reading and writing to text files, but the same principle can be applied to other sources of text. There are also IO classes that handle binary data, but we will not cover this.

Competency 430.1.2: Algorithm Design and Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

Text Files

Java provides over 30 classes to handle IO streams. This section introduces you to a few that read and write to text files, including two classes Java 5.0 introduced that make reading and writing to text files easy: scanner and formatter. File input and output is one of the most difficult concepts. You will have to understand this well to complete one of the performance tasks at the end of this course of study. It is strongly recommended that you complete all of the activities in this section to build solid competence in this topic.

As you complete the readings and tasks for this topic, ask yourself these questions:

- When would I use a FileReader?
- When would I use a FileWriter?
- What classes are available to read from and write to a text file?

□ Chapter 11: “Input/Output and Exception Handling”

Read chapter 11, section 11.1 (“Reading and Writing Text Files”). Code the examples in section 11.1 to get experience with reading and writing text files. Complete exercise P11.3 to get experience reading and writing files.

□ Research IO Classes in Java API Documentation

Review the following classes in the documentation. Note the constructors, methods, and attributes for each of the following IO classes:

- Scanner
 - Formatter
 - FileReader
 - FileWriter
 - PrintWriter
 - BufferedReader
 - BufferedWriter
-
- What is the syntax for constructing an object for each of the above IO classes?
 - Which class has methods that parse input from a file? In other words, which lets you read individual tokens from a file?
 - Which class reads an entire line from a file? Do you know how to tokenize a line of text read in?
 - How would you parse a line read in using BufferedReader?

□ Develop an Application That Uses Formatter and Scanner

Write a program that creates a class called DVDLogger with methods:

storeDVDInfo():

1. Continuously prompts the user to enter DVD title and price

2. Writes DVD title and price information to a text file using Formatter

readDVDInfo():

1. Opens the text file with the DVD information
2. Uses a scanner to read the DVD information
3. Prints text file contents to the screen

You will need to create a driver or test application that calls storeDVDInfo() and readDVDInfo().

□ Develop an Application That Uses FileReader and FileWriter

Write a program that creates a class called UnbufferedConverter with a method for the class UnbufferedConverter called unbufferedCaseConverter() and does the following:

1. Accepts a String argument called filename
2. Implements the method unbufferedCaseConverter() to use the FileReader class to read the contents of the text file
3. Converts all characters of the text file to uppercase
4. Uses the FileWriter class to write the converted contents to a new file called "unbufferedcaps.txt"

Week 9

Algorithm Development Using a Graphical User Interface

In this lesson you will learn how to create a graphical user interface (GUI), to build applications that are more user friendly. Interfaces that are intuitive, easy to use, and provide readable results positively impact satisfaction of an application.

Competency 430.1.2: Algorithm Design and Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

Javax.swing and Java.awt

In this section you will learn how to use javax.swing components to build graphical user interfaces. This topic is typically one of the most enjoyable to learn, and you will experience the ease in which a very sophisticated user interface can be developed.

□ Chapter 18: "Graphical User Interfaces"

As you complete the reading, pay attention to how the layout manager provides a few different layouts of the GUI components. Also pay close attention to the examples that use:

- JButton
 - JLabel
 - JRadioButton
 - JTextField
-
- What is the difference between javax.swing and java.awt?

□ Using the GUI Builder in NetBeans

URLs: <http://www.netbeans.org/kb/trails/matisse.html>

<http://www.netbeans.org/kb/60/java/quickstart-gui.html>
<http://www.netbeans.org/kb/60/java/quickstart-gui-legend.html>

Since you have downloaded NetBeans, you have a much easier and flexible way to create GUIs. You will need to create a GUI for one of your performance tasks. Please complete the tutorial provided in NetBeans to learn how to use the NetBeans GUI Builder using the links above. Each tutorial takes about 15 minutes.

Week 10

Performance Task 1 in TaskStream

You are now ready to complete the first performance task in TaskStream, a high stakes assessment where you will demonstrate all competencies developed in this course of study. Please log in to TaskStream and follow the instructions for completing and submitting the task.

Competency 430.1.1: Control Structures

The graduate develops working programs that use appropriate control structures and accurately evaluates paths in program code.

Competency 430.1.2: Algorithm Design and Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

Task 1

You are now ready to complete the first performance task in TaskStream, a high stakes assessment where you will demonstrate all competencies developed in this course of study. Please log in to TaskStream and follow the instructions for completing and submitting the task.

Performance Task 1

Refer for the AAT1 performance assessment through your mentor. Log in to TaskStream and follow the instructions for completing and submitting the first task.

Week 11

Performance Task 2 in TaskStream

You are now ready to complete the second performance task in TaskStream, a high stakes assessment where you will demonstrate all competencies developed in this course of study. Please log in to TaskStream and follow the instructions for completing and submitting the task.

Competency 430.1.1: Control Structures

The graduate develops working programs that use appropriate control structures and accurately evaluates paths in program code.

Competency 430.1.2: Algorithm Design & Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

Task 2

You are now ready to complete the first performance task in TaskStream, a high stakes assessment where you will demonstrate all competencies developed in this course of study. Please log in to TaskStream and follow the instructions for completing and submitting the task.

□ Performance Task 2

Refer for the AAT1 performance assessment through your mentor. Log in to TaskStream and follow the instructions for completing and submitting the second task.

Conclusion

Congratulations on successfully completing the activities in this course of study! You are now prepared to develop robust and user-friendly applications for a variety of industries using the Java programming language. Your applications will be object-oriented, making them maintainable and extensible. This is important to maintain customer satisfaction and demand for your applications.

Review of Major Points: Competencies for AAT1

As indicated in the introduction, this course of study covered the three competencies needed for successful completion of AAT1. Remember that control structures are key to development of application logic that causes certain behaviors and events to occur based on certain conditions. Many applications are developed to handle repetitive tasks for which your iteration control structures will be useful. Today's applications are typically data intensive; therefore having solid understanding of the characteristics of common data structures and good command of the collections provided by the Java API will be important to your future application development. The three competencies and their associated portions of the learning resources are listed below.

Competency 430.1.1: Control Structures

The graduate develops working programs that use appropriate control structures and accurately evaluates paths in program code.

- *Big Java* textbook chapter 5 (“Decisions”)
- *Big Java* textbook chapter 6 (“Iteration”)
- *Big Java* textbook chapter 11 (“Input/Output and Exception Handling”)

Competency 430.1.2: Algorithm Design and Development

The graduate designs and develops algorithms for problem solving and implements those algorithms using appropriate program code.

- *Big Java* textbook chapter 1 (“Introduction”)
- *Big Java* textbook chapter 4 (“Fundamental Data Types”)
- *Big Java* textbook chapter 18 (“Graphical User Interfaces”)

Competency 430.1.3: Use of Data Structures

The graduate develops working programs that use data structures for problem solving.

- *Big Java* textbook chapter 2 (“Using Objects”)
- *Big Java* textbook chapter 3 (“Implementing Classes”)
- *Big Java* textbook chapter 7 (“Array and Array Lists”)
- *Big Java* textbook chapter 8 (“Designing Classes”)

Transfer/Application to Work: How Will You Apply This Knowledge?

Successful completion of the AAT1 assessment validates that you have the basic hands-on skills and knowledge that software developers expect to understand and use. Skills include basic knowledge of the Java syntax and the Java API, program flow control structures, and data structures. You have also demonstrated that you can develop software applications that meet specified requirements. The knowledge and skill base that are acquired to pass this assessment are valuable for individuals working in software design and development.

You have learned the Java programming language and how to elicit requirements from a high-level application description. You developed the ability to design and develop an application that meets those requirements. You learned how to build a graphical user interface to create user friendly applications. You learned how to create robust code that validates user input and handles invalid entries. You learned how to review the Java API Documentation and use those well-designed classes and interfaces when appropriate.

These valuable skills prepare you to design, develop, and maintain software applications for a wide range of industries.

Next Steps: Passing AAT1 and Moving Onto ABT1

Once you have successfully completed all the tasks associated with the AAT1 competencies (readings, activities, COS tasks, and TaskStream performance tasks), you will receive a “PASS” on your AAP for the Introduction to Programming I (AAT1) assessment. You are now ready to begin working on Introduction to Programming II (ABT1) assessment. For ABT1, you will study object-oriented programming strategies and learn how to model designs using the Unified Modeling Language (UML). You will develop the ability to document your design and develop a test plan that will effectively validate and verify your application. Good luck!

Feedback

If you wish to provide feedback on this Course of Study, please contact Cheryl Bagshaw at cbagshaw@wgu.edu.