



This course supports the assessment for Scripting and Programming – Applications. The course covers 4 competencies and represents 4 competency units.

## Introduction

### Overview

Advancements in technology are made possible by programming a computer to do specialized tasks. A computer programming language is a simple language that breaks a specialized task into smaller sub-tasks. This allows the computer to achieve what we want it to by executing those smaller sub-tasks. Java is one of the most popular programming languages due to its simplicity, productivity, and performance at the enterprise application level. There are many Integrated Development Environments (IDEs) for writing Java Code, for example, Eclipse, BlueJ, Netbeans, etc. The IDE is for writing Java Code like a word processor would be for creating a document, for example, you might use Word, or Word Perfect, or MS Works, Google Docs., etc. This course introduces the use of the IDE BlueJ. BlueJ is a great IDE for beginning users, although it isn't necessarily the industry standard. If you plan on being a software developer, you might want to begin learning Eclipse (free resources will be provided for you within this course), but know that for this assessment, you may use any IDE you wish to complete the tasks (including BlueJ or Eclipse, or which you are most familiar with).

In this course you will explore the various aspects of the Java programming language by examining its syntax, the development environment where Java programs are written, and how to solve some real-world problems.

Watch the following video for an introduction to this course:

*Note: To download this video, right-click the following link and choose "Save as...": [download video](#).*

### Competencies

This course provides guidance to help you demonstrate the following 4 competencies:

- **Competency 4016.1.1: Introduction to Java Programming**  
The graduate defines programming languages, identifies common algorithms, and identifies the parts of the Java Programming Environment.
- **Competency 4016.1.2: Object-oriented Concepts**  
The graduate understands the object-oriented programming paradigm and identifies its elements.
- **Competency 4016.1.3: Variables and Data Types**  
The graduate declares, initializes, and assigns values to a variable and differentiates between primitive and object data types.
- **Competency 4016.1.4: Control Structures**  
The graduate utilizes decision and loop constructs to control the flow of a program.



## **Course Instructor Assistance**

As you prepare to successfully demonstrate competency in this subject, remember that course instructors stand ready to help you reach your educational goals. As subject matter experts, mentors enjoy and take pride in helping students become reflective learners, problem solvers, and critical thinkers. Course instructors are excited to hear from you and eager to work with you.

Successful students report that working with a course instructor is the key to their success. Course instructors are able to share tips on approaches, tools, and skills that can help you apply the content you're studying. They also provide guidance in assessment preparation strategies and troubleshoot areas of deficiency. Even if things don't work out on your first try, course instructors act as a support system to guide you through the revision process. You should expect to work with course instructors for the duration of your coursework, and you are encouraged to contact them as soon as you begin. Course instructors are fully committed to your success!

## **Preparing for Success**

The information in this section is provided to detail the resources available for you to use as you complete this course.

### **Learning Resources**

The learning resources listed in this section are required to complete the activities in this course. For many resources, WGU has provided automatic access through the course. However, you may need to manually enroll in or independently acquire other resources. Read the full instructions provided to ensure that you have access to all of your resources in a timely manner.

### **Getting Started with the Course**

Watch the following video for information on how to get started in the course:

*Note: To download this video, right-click the following link and choose "Save as...":* [download video](#).

### **Automatically Enrolled Resources**

You can access the learning resources listed in this section by clicking on the links provided throughout the course. You may be prompted to log in to the WGU student portal to access the resources.

### **WileyPLUS E-text**

The following textbook is available to you as an e-text within this course. You will be directly linked to the specific readings required within the activities that follow.

- Horstmann, C. (2010). *Java for everyone*. Hoboken, NJ: John Wiley & Sons. ISBN: 978-1-118-06331-6.

### **uDacity**

This is the main learning resource for this course and it contains most of the objectives and



exercises. You will be linked at the activity level to individual lessons in the following course:

- CS 046: Introduction to Programming in Java

### **Lynda.com**

This learning resource includes videos that will help supplement the other resources used in this course. You will be linked at the activity level to the following Lynda.com videos:

- Java Essential Training

### **Complete the Preassessment**

If you believe you have previous knowledge of some or all topics covered in this course, start by taking the preassessment before you begin and use its results to focus your studies.

- Complete the preassessment located in the Assessment tab.

Course instructors can help you develop a study plan based on your preassessment results.

### **Pacing Guide**

The pacing guide suggests a weekly structure to pace your completion of learning activities. It is provided as a suggestion and does not represent a mandatory schedule. Follow the pacing guide carefully to complete the course in the suggested time frame.

#### Week 1

- Introduction, and Fundamental Data Types

#### Week 2

- The String Data Type, and Control Structures

#### Week 3

- Looping Constructs, and Basic Classes and Objects

#### Week 4

- Data Structures, and Intermediate Object Oriented Topics

#### Week 5

- Advanced Object Oriented Concepts

#### Week 6

- Student Project



*Note: This pacing guide does not replace the course. Please continue to refer to the course for a comprehensive list of the resources and activities.*

## **Introduction to Java Programming**

This section will introduce programming languages, and how Java stands out as a very popular choice in today's world. This section will help you understand the concept of programming and set up the development environment that will be used throughout this course.

### **Java Essentials and Getting Started**

In this lesson, you will learn the basics of computer programming, and then dive into one of the most popular programming languages: Java. You will also set up the development environment and write simple Java programs using variables and methods.

This topic addresses the following competencies:

- **Competency 4016.1.1: Introduction to Java Programming**

The graduate defines programming languages, identifies common algorithms, and identifies the parts of the Java Programming Environment.

This topic highlights the following objectives:

- Define programming.
- Describe the Java programming language and its history.
- List commonly used IDEs.
- Explain the role of the JDK in the Java programming language.
- Write a program that outputs a string to the screen.
- Identify the features of a strongly typed language: variable declaration and type compatibility checking.
- Define variable and identify characteristics of variables.
- Initialize a variable.
- Identify the lifetime of a variable.
- Identify the different case conventions used in Java and the use of CamelCase for naming Java variables, methods, and classes.
- Discuss the importance of writing a test program (and eventually, unit tests).

#### **Watch: Lynda.com Lesson 1**

As you complete the material listed below, pay attention to the following key points:

- The role of the Java Virtual Machine
- There are multiple developing environments. Programmers can choose one according to their needs or preference.
- This course uses BlueJ for the exercises.

Watch the videos in the following Lynda.com lesson

- [lesson 1 \("What is Java"\)](#)



### **A Note About IDEs:**

This course utilizes BlueJ for exercises. BlueJ is a simple, easy-to-learn IDE for those who want an introduction to programming. If you are interested in working with an IDE more widely used in the software community, you are encouraged to complete the entire [Java Essential Training](#) course, which focuses on the IDE, Eclipse. Videos, as well as exercises are included in the Lynda.com course. If you plan to make a career of software development or your degree plan includes several other programming courses, you are also encouraged to use Eclipse to complete all hands-on labs as well as the final assessment for this course.

### **Watch: uDacity Lesson 1**

As you complete the listed material below, pay attention to the following key points:

- Importance of Java, and why it is the language of choice for so many developers.
- Installation of BlueJ (a Java IDE).
- Creating simple programs in Java.

Complete the following Udacity lesson:

- [lesson 1 \("Introduction"\)](#)

### **Read: WileyPlus chapters 1 and 2**

You can read more on this topic in the following sections of the *Java for Everyone* e-text:

- [chapter 1 \("Introduction"\)](#)
- [chapter 2 \("Fundamental Data Types"\)](#) (Skip the section labeled, "Reading Input.")

### **Watch: uDacity Lesson 2**

As you complete the listed material below, pay attention to the following key points:

- Understanding what objects are in a programming language.
- Understanding classes and methods.
- Understanding how to write and mutate variables.
- Understanding accessors and mutators.

Complete the following Udacity lesson up to the section labeled, "String Methods":

- [lesson 2 \("Objects"\)](#)

### **Read: WileyPlus Chapters 5 and 7**

You can read more on this topic in the following sections of the *Java for Everyone* e-text:



- [chapter 5 \(“Methods”\)](#)
- [chapter 7 \(“Input/Output and Exception Handling”\)](#) (Skip the section labeled, “Reading Input.”)

## Check Your Progress: uDacity Exercises

If you didn’t complete the exercises during the lesson, complete the following now:

1. [What do these strings print?](#)
2. [Udacity in a box](#)
3. [Block Tower](#)

## Object-Oriented Concepts

Each programming language adopts one of various paradigms to define its language structure. There are functional languages, procedural languages, and Object-oriented languages. Java falls into this last category. In this section you will learn what it means for a programming language to be an Object-oriented language.

### Classes and Objects

Object-oriented programming languages are designed around the concept of objects and their blueprints (classes). This section will introduce you to the concept of classes and objects and how we tend to think of things around us as objects. This is what makes object-oriented programming languages, such as Java, easier to comprehend and relate to from real-world examples.

This topic addresses the following competency:

- **Competency 4016.1.2: Object-oriented Concepts**  
The graduate understands the object-oriented programming paradigm and identifies its elements.

This topic highlights the following objectives:

- Define the concepts of class and object in Java.
- Create and instantiate a basic Java class.
- Create and use accessor and mutator methods.
- Generate a javadoc in a class, its variables, constructors, and methods.
- Create and use a method that takes parameters and returns a value.
- Create and use a constructor method.
- Create and use a method that uses the void-return type.
- Differentiate between passing arguments by value and by reference.
- Compare passing a primitive type and an object as a parameter.
- Create and use local and instance variables.

### Watch: Lesson 2 Objects



If you didn't complete lesson 2 in the previous section of this course of study, do it now.

As you complete the listed material below, pay attention to the following key points:

- Reflect on the concepts of object and class and their relationship to each other
- Interfaces are a powerful tool when working in a collaborative environment
- In programming a method is the ability [of an object] to do something rather than the way it is done
- Accessor and mutator methods make encapsulation possible

Complete the following Udacity lesson:

- [lesson 2 \("Objects"\)](#)

### **Watch: Lesson 3 Classes**

As you complete the listed material below, pay attention to the following key points:

- Writing a custom class
- Creating constructors
- Creating instance and local variables
- Dealing with programming errors

Complete the following Udacity lesson:

- [lesson 3 \("Classes"\)](#)

### **Read: WileyPlus Chapters 5 and 7**

You can read more on this topic in the following sections of the *Java for Everyone* e-text:

*Notice that you might have already read these chapters in previous sections of this course of study*

- [chapter 5 \("Methods"\)](#)
- [chapter 7 \("Input/Output and Exception Handling"\)](#) Skip the section labeled, "Reading Input."

## **Data Structures**

Data is the most important component of a computer program. Without input data, and generated output results (data), a program is not very useful. Data comes in many forms and shapes, and thus, a programming language such as Java needs to provide many types of place



holders for various types of data. In this lesson, you will learn about the two most powerful data structures in Java: arrays and array lists.

This topic addresses the following competency:

- **Competency 4016.1.2: Object-oriented Concepts**

The graduate understands the object-oriented programming paradigm and identifies its elements.

This topic highlights the following objectives:

- Create and use an array and identify its features.
- Create and use an ArrayList and identify its features.
- Compare array and ArrayList.
- Access, append, and modify an element from an ArrayList by position.
- Iterate through an array and an ArrayList within a control structure.

**Watch: Lesson 7 Array Lists and Arrays**

As you complete the listed material below, pay attention to the following key points:

- Understanding arrays and array lists
- Writing simple algorithms using arrays

Complete the following Udacity lesson:

- [lesson 7 \(“Array Lists and Arrays”\)](#)

**Watch Lesson 8: More About Classes**

As you complete the listed material below, pay attention to the following key points:

- Writing code that distinguishes static versus instance methods
- Writing code that distinguishes static versus instance variables
- Understanding packages

Complete the following Udacity lesson:

- [lesson 8 \(“More About Classes”\)](#)

**Read: WileyPlus Chapter 6**

You can read more on this topic in the following section of the *Java for Everyone* e-text:

- [chapter 6 \(“Arrays and Array Lists”\)](#)

**Read: WileyPlus Chapter 8**

You can read more on this topic in the following section of the *Java for Everyone* e-text:





- [chapter 8 \(Objects and Classes\)](#)

## Advanced Object-oriented Concepts

In this section, you will look at the ability of Java as an object-oriented language, Java to implement inheritance of properties, as well as to provide polymorphic behavior. You will also learn the concept of *encapsulation* although we have been using it in previous lessons.

This topic addresses the following competencies:

- **Competency 4016.1.2: Object-oriented Concepts**  
The graduate understands the object-oriented programming paradigm and identifies its elements.

This topic highlights the following objectives:

- Explain inheritance and identify its uses.
- Discuss polymorphism.
- Discuss encapsulation.
- Use the *this* keyword in a class method.

### Watch: Lesson 9 Interfaces and Inheritance

As you complete the listed material below, pay attention to the following key points:

- Writing interfaces and their implementations
- What is polymorphism?
- Understanding “instanceof” and casting
- Understanding the Comparable interface
- Implementing subclass methods

Complete the following Udacity lesson:

- [lesson 9 \(“Interfaces and Inheritance”\)](#)

### Read: WileyPlus Chapter 9

You can read more on this topic in the following section of the *Java for Everyone* e-text:

- [chapter 9 \(“Inheritance and Interfaces”\)](#)

## Variables and Data Types

Every language has its simple words. In this section you will learn the basic keywords that are used in Java, and the various types that are available to allow encapsulation of various data types. You will also learn how data within the same data type interact with each other through the use of operators.

### Primitive/Fundamental Data Types

This topic will help you understand the structure of the Java language and its development environment.



This topic addresses the following competency:

- **Competency 4016.1.3: Variables and Data Types**

The graduate declares, initializes, and assigns values to a variable and differentiates between primitive and object data types.

This topic highlights the following objectives:

- Identify different data types, their characteristics, and common applications.
- Evaluate arithmetic operations on different numeric types (e.g. int, double, char).
- Demonstrate understanding of operator precedence.
- Use a cast operator on a primitive data type.
- Identify code where a type mismatch compiler error would be raised.
- Predict the output of a simple Boolean operation.

**Watch: Lesson 4 Fundamental Data Types**

As you complete the listed material below, pay attention to the following key points:

- Becoming familiar with Java defined numerical classes

Complete the following Udacity lesson:

- [lesson 4 \("Fundamental Data Types"\)](#)

The following sections of this lesson are optional:

- Reading Input
- Your Own Sunset Effect
- Formatted Output
- Practice Formatted Output
- Formatted Output Practice Part 2

**Read: WileyPlus chapters 2 and 3**

You can read more on this topic in the following section of the *Java for Everyone* e-text.

- [chapter 2 \("Fundamental Data Types"\)](#)
- [chapter 3 \("Decisions"\)](#)

*Note: You might have already read these chapters in previous sections of this course.*

## **The String Data Type**

Most data available can be handled as strings or text, which makes this class extra special. There are many pre-defined methods available to Java programmers that are used to create, manipulate, and process text. This will introduce you to the various methods that are already implemented and that save millions of developers a lot of coding time.



This topic addresses the following competency:

- **Competency 4016.1.3: Variables and Data Types**

The graduate declares, initializes, and assigns values to a variable and differentiates between primitive and object data types.

This topic highlights the following objectives:

- Describe the string as an immutable type.
- Identify the string property that contains its length.
- Describe chained or cascading method calls.
- Concatenate strings.
- Identify the difference between indexOf and substring.
- Use the Scanner class to read in numeric data from the console.

**Watch: Lesson 2 Objects**

As you complete the listed material below, pay attention to the following key points:

- Understanding String methods

Complete the following Udacity lesson:

- [lesson 2 \(“Objects”\)](#)

**Watch: Lesson 4 Fundamental Data Types**

As you complete the listed material below, pay attention to the following key points:

- Understanding the Scanner class.
- Understanding how to format numeric data.

Complete the following Udacity lesson:

- [lesson 4 \(“Fundamental Data Types”\)](#)

The following sections of this lesson are optional:

- Reading Input
- Your Own Sunset Effect
- Formatted Output
- Practice Formatted Output
- Formatted Output Practice Part 2

**Read: WileyPlus chapters 2 and 5**

You can read more on this topic in the following section of the *Java for Everyone* e-text:

*Notice that you might have already read these chapters in previous sections of this course*



- [chapter 2 \(“Fundamental Data Types”\)](#)
- [chapter 5 \(“Methods”\)](#)

*Note: You might have already read these chapters in previous sections of this course.*

## Control Structures

Programming in general requires basic control constructs in order to allow efficient code writing. Instead of writing two different versions of a code based on what the input data is, one can write a decision statement. A *decisions* statement will choose one code path based on the input. In this section you will learn some very powerful tools available to any programming language, including Java that would provide the ability to control code paths within a program.

### Decisions

Choosing one of two possible decisions is a very common problem that a programmer will deal with. Programming languages are no different. They provide a way to check for the value of some data, and based on this value, the program will execute one section of the code or another. In this topic you will learn about IF/ELSE constructs.

This topic addresses the following competency:

- **Competency 4016.1.4: Control Structures**  
The graduate utilizes decision and loop constructs to control the flow of a program.

This topic highlights the following objectives:

- Use an if-else statement and identify its parts.
- Understand relational operators.
- Differentiate between *empty string*, *zero*, and *null*.
- Use compound conditions involving the && (and), || (or), and ! (not) operators.

### Watch: Lesson 5 Decisions

As you complete the listed material below, pay attention to the following key points:

- Understanding if-else statements
- Understanding conditional operators

Complete the following Udacity lesson:

- [lesson 5 \(“Decisions”\)](#)

### Read: WileyPlus Chapter 3

You can read more on this topic in the following section of the *Java for Everyone* e-text:

- [chapter 3 \(“Decisions”\)](#)

*Note: You might have already read this chapter in previous sections of this course.*



## Looping Constructs

Looping is another concept available in every programming language. Instead of having to duplicate a code to run it multiple times, Java provides the ability to loop through a section of code. In this section you will learn about three different types of looping tools in Java: FOR, WHILE, and DO loops. You will also learn the difference among them, and how to use all of them effectively to write any powerful program.

This topic addresses the following competency:

- **Competency 4016.1.4: Control Structures**

The graduate utilizes decision and loop constructs to control the flow of a program.

This topic highlights the following objectives:

- Identify uses for looping constructs
- Recognize an infinite loop and explain its effects in a program
- Use loops in a simple program

### Watch: Lesson 6 Loops

As you complete the listed material below, pay attention to the following key points:

- Understanding While loops
- Understanding Do loops
- Pay attention to the difference between Do loops which always execute at least once and While loops which might never execute if the condition is not met.
- Understanding For loops
- Understanding infinite loops, and avoiding them

Complete the following Udacity lesson:

- [lesson 6 \("Loops"\)](#)

### Read: WileyPlus Chapter 4

You can read more on this topic in the following section of the *Java for Everyone* e-text:

- [chapter 4 \("Loops"\)](#) (Skip the section labeled, "Application: Processing Sentinel Values.")

*Note: You might have already read this chapter in previous sections of this course.*

### Complete: Scripting & Programming ? Apps VGT1 Student Project

Complete the following student project in [TaskStream](#):

- VGT1: Scripting & Programming - Apps



For details about this performance assessment, see the "Assessment" tab in this course.

## Final Steps

Congratulations on completing the activities in this course! This course has prepared you to complete the assessment associated with this course. If you have not already been directed to complete it, schedule and complete the assessment now.

### Complete the Preassessment

The preassessment can help you determine your level of preparation. It is highly recommended that you pass the preassessment before attempting the final exam.

- Complete the preassessment located in the Assessment tab.

Course instructors can help you develop a review plan based on your preassessment results.

### First Attempt Checklist

One of the many things that makes WGU unique is its competency-based education model. If you know the material, all you have to do is prove it by passing the exam. If you can do this, you can accelerate the receipt of your degree.

To make sure you have the best chance possible to pass the exam on your first attempt, the following steps should be completed successfully before you take it:

1. Complete all the assigned lessons in the Udacity online course, CS046, Introduction to Programming.
2. Complete all of the programming exercises when requested in the interactive Udacity curriculum. When complete, save a screenshot of your progress, which is available from the "Progress" tab. Please email this screenshot to [cmprogramming@wgu.edu](mailto:cmprogramming@wgu.edu) with subject: "C169 – Progress Report – Student: YOUR NAME" as a record of your engagement in the course.
3. To practice for the DQC1 assessment, use the secondary learning resource WileyPlus, *Java for Everyone*. Complete the "Horstmann's Drill" and "Practice Picks" for chapters 1-9 in WileyPlus, striving for 80%+ on the first attempt for each. If you discover topics that require review, you can click directly into the relevant section of *Java for Everyone* from any "Horstmann's Drill" and "Practice Picks" question.
4. Take the PDQC pre-assessment for the course, striving for 85%+. Use the coaching report to determine where you still need to review (Score 70%-) and reach out to the course instructors with any questions you have.

If you have completed the steps above and you feel comfortable with all of the concepts presented, you are most likely ready to attempt the exam.

If you fail your first attempt, you will be required to contact the course instructor to see what went wrong and how you can prepare to ensure a successful second attempt. After determining you are ready, your course instructor will approve your request once to make another exam attempt.