Hello, and welcome!

Computers play a major role in our world and daily lives—from smartphones to laptops to gaming systems and more. And they range from simple to complex in their operations and abilities. But there is a common connection among them all—scripting and programming. And whether you are learning to build a house, paint a picture, or work in information technology—it all starts with a solid foundation. In this course, you will examine the fundamental principles of scripting and programming. And you will gain the necessary skills to advance your career to the next level.

We invite you to take a minute to learn about the course by reviewing the following information. This way, you will be better able to understand the expectations of the course as a whole. Then you can determine how to manage your time and efforts as you navigate through it.

You are in the right place. You belong here. You can do this!
Course Description and Competencies

What to Expect

In this course, you will examine the fundamental principles of programming. You will look at variables, data types, flow control, and design concepts. You will survey various languages used in scripting. And you will also gain an understanding of the logic and outcome of simple algorithms.

Throughout this course, you will find a variety of interactive elements. These include an interactive textbook, participation activities, and labs. Engaging with these activities reinforces new learning. It also complements the knowledge you bring to this area of study.

The objective assessment allows you to demonstrate three core competencies from the course. You will have two attempts to pass the exam.

- 1 final assessment
- 3 competency units

This course covers the following competencies:

1. The learner identifies scripts for computer program requirements.
2. The learner uses fundamental programming elements as part of common computer programming tasks.
3. The learner explains the logic and outcome of simple algorithms.

Course Outline

<table>
<thead>
<tr>
<th>Units</th>
<th>Upon completion of this unit, you will be able to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>A. Define what a computer program is and what they do</td>
</tr>
<tr>
<td></td>
<td>B. Describe the purpose of a flowchart</td>
</tr>
<tr>
<td></td>
<td>C. Explain how a computer program represents data</td>
</tr>
<tr>
<td></td>
<td>D. Describe the purpose of pseudocode</td>
</tr>
<tr>
<td>Variables &amp; Assignments</td>
<td>A. Explain the purpose of variables, assignment statements, and expressions</td>
</tr>
<tr>
<td></td>
<td>B. Follow identifier naming conventions</td>
</tr>
<tr>
<td></td>
<td>C. Describe how to choose a variable's type</td>
</tr>
<tr>
<td></td>
<td>D. Describe functions and type conversions</td>
</tr>
</tbody>
</table>
| Branches | A. Define a branch in a flowchart  
B. Explain a decision in a flowchart  
C. Describe if-else statements  
D. Explain nested branches |
|-----------|---------------------------------------------------------------------|
| Loops    | A. Describe a loop and how one is implemented in a program  
B. Explain an infinite loop and a nested loop  
C. Compare while loops, for loops, and do loops  
D. Build complex Cascading Style Sheets (CSS) |
| Arrays   | A. Define an array  
B. Describe an array index  
C. Define an array variable  
D. Explain how to iterate through an array |
| User-Defined Functions | A. Define the purpose of a function  
B. Describe the role of a parameter in a function  
C. Explain modular development  
D. Explain incremental development |
| Algorithms | A. Define what an algorithm is  
B. Explain how to measure an algorithm's efficiency  
C. Describe practical applications of algorithms  
D. Compare linear searches and binary searches |
| The Design Process | A. Describe the system development life cycle (SDLC)  
B. Explain the difference between Agile and Waterfall  
C. Define the purpose of an object in programming  
D. Define the Universal Modeling Language (UML) |
| Software Topics | A. Explain the difference between a compiled language and an interpreted language  
B. Compare statically typed and dynamically typed languages  
C. Describe object-oriented languages and markup languages  
D. Explain how libraries improve productivity in programming |
| Troubleshooting | A. Explain the troubleshooting process  
B. Define the role of a hypothesis in troubleshooting  
C. Explain an asymmetric test  
D. Describe a hierarchical hypothesis |
| Debugging | A. Describe the purpose of debugging in programming  
B. Explain debugging output statements  
C. Describe hierarchical debugging |
Technology Requirements

We want to be sure you have the tools to succeed! Review the Computer System and Technology Requirements to learn about the technology you'll need. If you have questions about your setup, contact support@academy.wgu.edu.

Key Contacts

Your Fellow Learners

Check out the Scripting and Programming -- Foundations Lobby in the course site! In this online community, you can ask questions and explore ideas. You can connect with your fellow learners. When you use this site, you will realize that other learners may have the same questions you have. You can all benefit from learning together!

Technical Support

If you encounter technical issues, be sure to contact the Help Desk. Just submit a Support Request for assistance.

Program Support

Do you have questions about your account? Our Academy Support Team has answers. They can help with billing, switching courses, and other requests. You can contact them at (888) 320-0540 or support@academy.wgu.edu.

Accommodations

WGU Academy provides compliant and accessible learning experiences. If you require accommodation, please contact us at the start of the course. You can email StudentAffairs@academy.wgu.edu or call (888) 320-0540. We are committed to ensuring that all students with disabilities have equal access to WGU Academy's services and materials. We strive to use best practices for accessibility. Our goal is to conform to existing U.S. laws. These include the Americans with Disabilities Act and Section 504 and Section 508 of the Rehabilitation Act. Our learning management system (LMS) platform is Open edX. Open edX's commitment to accessible content is published on their Website Accessibility Policy.