

# Introduction to Programming 1B: Problem Solving Through Programming

Building on the prior prerequisite course, discover how programming can solve a vast array of problems! Plan and develop a problem-solving program while performing testing, debugging, and quality assurance procedures. Design and plan your own app as part of your capstone project to give you a thorough introduction to the world of programming.

## Unit 1: Designing Programs

Producing a quality piece of software happens through careful planning and preparation. Determining the hardware and software requirements, setting up a database structure, and creating a data flow diagram are all important parts of the design process. In this unit, we will learn how the software development life cycle drives the creation of software. We will also create a software design document that provides necessary information about the software product. Get ready to dive into the documentation world!

### What will you learn in this unit?

- Write a software design document
- Compare and contrast structured programming and object-oriented approaches
- Explain the details of how a program is executed
- Describe different types of documentation used in the software development life cycle

## Unit 2: Plan for Success

We already know how important it is to plan before developing a program. But how exactly do we design a program before we write it, especially if it is a large piece of software? In this unit, we will explore various tools like diagrams and version control systems that will help the programming team plan for success. We will create a data flow diagram, a process logic diagram, pseudocode, and a flowchart in order to get a hands-on perspective. We will end by converting some of our designs to a block-based programming language.

### What will you learn in this unit?

- Describe and create tools for designing a program
- Understand and explain how version control systems work
- Determine the flow of data through a program
- Convert pseudocode and flowcharts to programs

## Unit 3: Abstraction

Do you know how exactly a car works when you turn the engine on and drive? Do you know all the details of how your refrigerator keeps your food cold? What about a washing machine? How do the dials or the interface communicate with the machine to get your clothes clean? Thankfully, we can use these machines and appliances without being mechanical engineers, thanks to the concept of abstraction, which allows us to hide complex details. In this unit, we will explore abstraction from various angles and apply it to computer hardware and programming languages. We will end with some fun abstraction using Scratch.

### What will you learn in this unit?

- Understand and explain how models and simulations use abstraction to raise and answer questions
- Use global and local variables in Python
- Explain how multiple levels of abstraction are used in computers

- Provide specific examples of how models and simulations are used in the real world

## **Unit 4: Quality Assurance**

When people are in the process of buying a house, one step that should never be skipped is having a home inspector check the house over. Home inspectors have a list of areas they check to make sure everything is up to code. They examine things such as the plumbing and electrical work, the windows, and the roof. Once they have completed their inspection, they write up a report and suggest areas that may need improvement before the buyers finalize their purchase. Similarly, Quality assurance (QA) specialists make sure that software meets certain regulations and standards. Like home inspectors, they also work with a long checklist of areas to ensure that the software meets expectations. The QA process touches most of the software development life cycle, so get ready to view software through the lens of an enforcer, auditor, maintainer, and documenter!

### **What will you learn in this unit?**

- Explain the software quality assurance process
- Follow quality assurance procedures for testing, identifying problems, and tracking resolutions
- Maintain programs by updating documentation
- Provide user documentation and support

## **Introduction to Programming 1b Midterm Exam**

- Review information acquired and mastered from this course up to this point.
- Take a course exam based on material from the first four units in this course (Note: You will be able to open this exam only one time.)

## **Unit 5: Running the Tests**

Software bugs can be serious and even detrimental to the project if not detected. In 1998, the Mars Climate Orbiter was launched into space but was destroyed upon entering Mars' atmosphere. The probe's software used imperial measurements instead of metric units as expected, which was a serious bug that caused a fatal defect. Testing is an important part of the software development life cycle that can mitigate serious flaws. Read on to learn the ins and outs of testing!

### **What will you learn in this unit?**

- Identify and debug errors
- Explain differences between types of testing done throughout the software development life cycle
- Create robust programs that handle errors effectively
- Create a test plan

## **Unit 6: Legal and Ethical Computing**

Breaking the law can have serious consequences in the computing industry. But there are many ethical dilemmas and gray areas that have not been addressed by the law, especially as technology advances. How will driverless cars choose who to prioritize—pedestrians or passengers? What kind of consumer data can companies sell? If there is a loophole in an acceptable use policy is it ok to exploit it? These questions require a careful look at ethics and some good guiding principles.

### **What will you learn in this unit?**

- Describe differences between types of software licensing
- Analyze the impact of computer laws on privacy and security
- Explain the ethical responsibilities of a computer professional
- Evaluate issues relating to copyright and plagiarism

## **Unit 7: Safe and Secure**

Our world is full of risks and threats that can come from a variety of sources. Natural disasters, vandalism, and power outages can wreak havoc on IT facilities. Data communication systems can be hacked, and software applications can be vulnerable if their code is not robust. Stay tuned to see how we can prevent these and other risks. We will also check out how the internet functions so that we can better understand how to prevent cybersecurity risks.

### **What will you learn in this unit?**

- Identify risks to IT facilities, data communication systems, and software applications
- Identify risk recovery steps for various situations
- Discuss privacy and security issues that are associated with the internet
- Explain the various risk response strategies and when they should be used

## **Unit 8: Skill Spotlight: Involved and Informed**

Did you know there are over 4 billion internet users in the world? That means data is being generated at an unprecedented level. How do we keep track of meaningful information in the workplace so that it is useful and effective? We are going to explore personal information management (PIM) applications and get some firsthand experience using several. We will also dive into the world of technical writing by learning how to write a computer research report. And we will end by considering how we can use our skills now and in the future to improve the world!

### **What will you learn in this unit?**

- Use personal information management (PIM) applications to increase workplace efficiency
- Conduct technical research to gather information for a research report
- Identify and document performance goals and monitor progress toward those goals
- Identify ways to use technology to support collaboration and lifelong learning

## **Introduction to Programming 1b Final Exam**

- Review information acquired and mastered from this course up to this point.
- Take a course exam based on material from units five to eight in this course – the last four units. (Note: You will be able to open this exam only one time.)