

# PLATO Course Computer Programming I B

## Course Overview

This one-semester course is intended as a practical, hands-on guide to help you understand various phases of the software development life cycle (SDLC). This course has four Units with 14 lessons and four Unit Activities. Each lesson contains one or more Lesson Activities.

This course describes various phases of the SDLC such as analysis, design, development, testing, and implementation. This course describes software development methodologies, various types of project plans, Unified Modeling Language (UML) design, various types of testing, and system implementation. This course also identifies various security threats and risks to computer systems and the methods to mitigate them.

You will submit the Unit Activity documents to your teacher, and you will grade your work in the Lesson Activities by comparing them with given sample responses. The Unit Activities (submitted to the teacher) and the Lesson Activities (self-checked) are the major components of this course. There are other assessment components, namely the mastery test questions that feature along with the lesson; the pre- and post-test questions that come at the beginning and end of the unit respectively, and an end-of-semester test. All of these tests are a combination of simple multiple-choice questions and technology enhanced (TE) questions.

## Course Goals

This course will help you meet the following goals:

- Identify the different phases of the SDLC.
- Describe and compare the different methods of software development.
- Describe the different types of plans created during software development.
- Create a design document using Unified Modeling Language (UML).
- Identify the different types of software testing.
- Describe how to implement a system in a production environment.
- Identify the different types of security risks and threats to computer systems.
- Identify the different methods to respond to security risks and threats.

## Prerequisite Skills

PLATO Course Computer Programming I B has a prerequisite course, Computer Programming I A. Also, these fundamental skills will be helpful:

- basic math knowledge
- ability to visualize and apply creativity and innovation
- general familiarity with the writing process and following guidelines

## General Skills

To participate in this course, you should be able to do the following:

- Perform basic operations, and use word processing, spreadsheet, and presentation software.
- Perform online research using various search engines and library databases.
- Communicate through email and participate in discussion boards.

*For a complete list of general skills that are required for participation in online courses, refer to the Prerequisites section of the Plato Student Orientation document, found at the beginning of this course.*

## Credit Value

PLATO Course Computer Programming I B is a 0.5-credit course.

## Course Materials

- Notebook
- Computer with Internet connection and speakers or headphones
- Microsoft Word or equivalent
- Microsoft Excel or equivalent
- Microsoft PowerPoint or equivalent

## Course Pacing Guide

This course description and pacing guide is intended to help you stay on schedule with your work. Note that your course instructor may modify the schedule to meet the specific needs of your class.

## Course Structure

### Unit 1: Software Development Life Cycle and Initial Phases

#### Summary

In this unit, you will identify the different phases of the software development life cycle (SDLC). You will describe various software development methodologies, such as the waterfall and spiral models. You will also compare various software development methodologies based on their advantages and disadvantages. Additionally in this unit, you will describe the methods of requirements gathering and the types of project plans.

Day	Activity/Objective	Type
1 day: 1	<b>Syllabus and Plato Student Orientation</b> <i>Review the Plato Student Orientation and Course Syllabus at the beginning of this course.</i>	Course Orientation
4 days: 2–5	<b>The Software Development Life Cycle</b> <i>Identify the different phases of the software development life cycle (SDLC).</i>	Lesson
4 days: 6–9	<b>Software Development Methodologies</b> <i>Describe and compare the different methods of software development, such as the waterfall method and the rapid prototyping method.</i>	Lesson
4 days: 10–13	<b>Requirements Gathering and Analysis</b> <i>Describe and analyze client and project requirements.</i>	Lesson
1 day: 14	<b>Space Jumble</b>	Game
4 days: 15–18	<b>Planning</b> <i>Describe various tasks in the planning phase of the SDLC.</i>	Lesson
5 days: 19–23	<b>Unit Activity/Threaded Discussion—Unit 1</b>	Unit Activity
1 day: 24	<b>Posttest—Unit 1</b>	Assessment

## Unit 2: Development Phases of the SDLC

### Summary

In this unit, you will describe various design methodologies and tools that you can use to design your software application. Design methods include object-oriented and structured methods, while tools include flowcharts and the Unified Modeling Language (UML). You will also create a software design using UML. In the last lesson of this unit, you will describe the different tasks involved in the coding phase of the SDLC.

Day	Activity/Objective	Type
4 days: 25–28	<b>Design Methodologies</b> <i>Identify the different methodologies and tools for developing a software design.</i>	Lesson
4 days: 29–32	<b>Unified Modeling Language</b> <i>Create a design document using Unified Modeling Language (UML).</i>	Lesson
1 day: 33	<b>Para Jumble</b>	Game
4 days: 34–37	<b>Coding</b> <i>Describe various tasks in the coding phase of the SDLC.</i>	Lesson
5 days: 38–42	<b>Unit Activity/Threaded Discussion —Unit 2</b>	Unit Activity
1 day: 43	<b>Posttest—Unit 2</b>	Assessment

## Unit 3: Testing and Quality Control

### Summary

In this unit, you will identify the different types of testing. You will identify the different types of programming errors, such as syntax errors and logical errors. You will describe the importance of test plans and test scripts. Finally, you will explain the role of quality assurance in improving the quality of software.

Day	Activity/Objective	Type
4 days: 44–47	<b>Testing</b> <i>Identify the different types of software testing.</i>	Lesson
4 days: 48–51	<b>Test Plan and Test Scripts</b> <i>Describe how to use a test plan and test scripts for testing.</i>	Lesson
1 day: 52	<b>Space Jumble</b>	Game
4 days: 53–56	<b>Quality Control</b> <i>Explain how quality control helps in improving processes and the quality of the software.</i>	Lesson
5 days: 57–61	<b>Unit Activity/Threaded Discussion —Unit 3</b>	Unit Activity
1 day: 62	<b>Post test- Unit 3</b>	

## Unit 4: Successful and Safe Project Implementation

### Summary

In this unit, you will describe how to implement a system in a production environment. You will describe the importance of maintenance in the SDLC. You will identify the different types of documentation used in SDLC. In the latter part of the unit, you will identify the different types of security risks and threats to your computer system and ways to mitigate and recover from them.

Day	Activity/Objective	Type
4 days: 63–66	<b>Implementation and Maintenance</b> <i>Describe how to implement a system in a production environment and how companies provide product maintenance for customers.</i>	Lesson
3 days: 67–69	<b>Types of Documentation</b> <i>Identify the different types of documentation used in the software development life cycle.</i>	Lesson
1 day: 70	<b>Para Jumble</b>	Game
3 days: 71–73	<b>Need for Computer Applications</b>	Course Activity
4 days: 74–77	<b>Information Security Risks and Threats</b> <i>Identify the different types of security risks and threats to computer systems.</i>	Lesson
4 days: 78–81	<b>Disaster Recovery</b> <i>Identify the different methods to respond to security risks and threats.</i>	Lesson
5 days: 82–86	<b>Unit Activity/Threaded Discussion —Unit 4</b>	Unit Activity
1 day: 87	<b>Space Jumble</b>	Game
1 day: 88	<b>Posttest—Unit 4</b>	Assessment
1 day: 89	<b>Semester Review</b>	
1 day: 90	<b>End-of-Semester Test</b>	Assessment