

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 teacher may modify the schedule to meet the specific needs of your class.

Course Components and Grading Rubric

The table gives a breakdown of the weight for each component in the course. Weight represents the percentage of the total score coming from each activity.

Course Components	Count	Weight
Pretest. <i>Pretests are optional assessments, typically designed for credit recovery use. If a student shows mastery of a lesson's objective, the student may be automatically exempted from that lesson in the upcoming unit. Typically, teachers do not choose to employ exemptive pretests for first-time credit courses. Pretests are not included as a component of the student's final grade.</i>	4	0%
Module. <i>Each module in this course contains an interactive tutorial and an associated mastery test. Tutorials may include one or more Lesson Activities that constitute tasks associated with the tutorial. The module score comes from a student's score on the mastery test.</i>	14	20%
Discussion. <i>Online discussions allow for higher-order thinking about terminal objectives. An online threaded discussion mirrors the educational experience of a classroom discussion. Teachers can initiate a discussion by asking a complex, open-ended question. Students can engage in the discussion by responding both to the question and to the thoughts of others. Each unit in a course has one predefined discussion topic; teachers may add more discussion topics.</i>	4	10%
Unit Activity. <i>Unit Activities are at the end a unit and constitute one or more small tasks. Their purpose is to deepen understanding of key unit concepts and tie them together. Each Unit Activity includes a simple rubric. The teacher versions include both a rubric and modeled sample answers. Unit Activities are teacher graded.</i>	4	20%
Posttest. <i>The posttest appears at the end of the unit and mirrors the pretest in structure, content, and complexity.</i>	4	20%
Course Activity. <i>Course Activities are similar to Unit Activities in scope but may be found at any point in the course, either to prepare the student for new learning or to act as a performance-based activity required for a learning objective. Like Unit Activities, Course Activities include simple rubrics, and sample answers are available for teachers. Course Activities are teacher graded.</i>	1	10%
End of Semester Test. <i>The end of semester test (EOS) appears at the end of the course. Students are delivered a few items from every tutorial in the course in order to assess the major course objectives.</i>	1	20%
Total	32	100%

**Teachers may manually adjust these weights if desired, per district grading requirements.*

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	Syllabus and Plato Student Orientation <i>Review the Plato Student Orientation and Course Syllabus at the beginning of this course.</i>	Course Orientation
4 days: 2–5	The Software Development Life Cycle <i>Identify the different phases of the software development life cycle (SDLC).</i>	Lesson
4 days: 6–9	Software Development Methodologies <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	Requirements Gathering and Analysis <i>Describe and analyze client and project requirements.</i>	Lesson
1 day: 14	Space Jumble	Game
4 days: 15–18	Planning <i>Describe various tasks in the planning phase of the SDLC.</i>	Lesson
5 days: 19–23	Unit Activity/Threaded Discussion—Unit 1	Unit Activity
1 day: 24	Posttest—Unit 1	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	Design Methodologies <i>Identify the different methodologies and tools for developing a software design.</i>	Lesson
4 days: 29–32	Unified Modeling Language <i>Create a design document using Unified Modeling Language (UML).</i>	Lesson
1 day: 33	Para Jumble	Game
4 days: 34–37	Coding <i>Describe various tasks in the coding phase of the SDLC.</i>	Lesson
5 days: 38–42	Unit Activity/Threaded Discussion —Unit 2	Unit Activity
1 day: 43	Posttest—Unit 2	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	Testing <i>Identify the different types of software testing.</i>	Lesson
4 days: 48–51	Test Plan and Test Scripts <i>Describe how to use a test plan and test scripts for testing.</i>	Lesson
1 day: 52	Space Jumble	Game
4 days: 53–56	Quality Control <i>Explain how quality control helps in improving processes and the quality of the software.</i>	Lesson
5 days: 57–61	Unit Activity/Threaded Discussion —Unit 3	Unit Activity
1 day: 62	Post test- Unit 3	

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	Implementation and Maintenance <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	Types of Documentation <i>Identify the different types of documentation used in the software development life cycle.</i>	Lesson
1 day: 70	Para Jumble	Game
3 days: 71–73	Need for Computer Applications	Course Activity
4 days: 74–77	Information Security Risks and Threats <i>Identify the different types of security risks and threats to computer systems.</i>	Lesson
4 days: 78–81	Disaster Recovery <i>Identify the different methods to respond to security risks and threats.</i>	Lesson
5 days: 82–86	Unit Activity/Threaded Discussion —Unit 4	Unit Activity
1 day: 87	Space Jumble	Game
1 day: 88	Posttest—Unit 4	Assessment
1 day: 89	Semester Review	
1 day: 90	End-of-Semester Test	Assessment

Course Map

You will achieve course level objectives by completing each lesson's instruction, assignments, and assessments. For a detailed look at how the materials meet these objectives, review the [course map for Semester B](#).