

Beginning Student Language Module

[Help](#)

This page provides a complete description of the material for the first module of the course including: topics and learning goals; descriptions of all videos; practice and homework problems; as well as specific tips for how to do well with this module's material. Each module of the course will have a similar page and you can access them all from the navigation bar to the left.












This focus of this module is learning the Beginning Student Language (BSL) that we will be using in Part 1 of Systematic Program Design. For students who have already programmed this module may seem too easy – but be careful! Be sure to learn **this programming language**. While this language forms the conceptual core of nearly every programming language you might have used, it is also different in important ways. Take the time to go through this material carefully.



































Students who have never programmed before -- and these students are the primary audience for the course -- this module may take more time. The videos this module are paced so that you can learn the basic mechanisms of the teaching language we will be using. The videos go through not just the language but also the tools we will be using. Please avoid the temptation to skim this material.

The material in this module should take **approximately 4-5 hours** of dedicated time to complete, including working along with the lecture videos, doing the practice problems, doing the homework problems and the short quiz.

Learning Goals







- Be able to write expressions that operate on primitive data including numbers, strings, images and booleans.
- Be able to write constant and function definitions.
- Be able to write out the step-by-step evaluation of simple expressions including function calls.
- Be able to use the stepper to automatically step through the evaluation of an expression.







Topic	Length (mm:ss)	Starter File	Downloads
BSL - Introduction Motivation and goals of the course. How and why this course differs from other introductory programming courses.	5:33	<i>none</i>	   
BSL - How to Study A introductory navigation through the course website, and some advice on studying.	6:52	<i>none</i>	   
BSL - Expressions DrRacket, numbers, forming expressions that operate on numbers. <i>NOTE: Before starting this lecture download the latest</i>	8:30	pythag-starter.rkt	  

NOTE: Before starting this lecture download the latest version of DrRacket , install the download on your computer and then start the DrRacket application.		starter.rkt	   
BSL - Evaluation Rules for evaluating primitive call expressions.	6:56	<i>none</i>	   
BSL - Strings and Images String and image values. Primitives that operate on strings and images.	11:28	<i>none</i>	   
BSL - Constant Definitions Constant definitions, forming constant definitions and rules for evaluating constants.	6:15	<i>none</i>	   
BSL - Pacing and Doing Well in the Course On the pacing of the course, and reminders of how to do well.	1:15	<i>none</i>	  
BSL - Function Definitions Function definitions, forming function definitions and rules for evaluating function calls.	11:01	function-definitions-starter.rkt	   
BSL - Booleans and if Expressions Boolean Values. Forming if expressions and rules for evaluating if expressions. This video is particularly long because it includes a detailed step-by-step evaluation example. You may want to skim that example and come back to it later.	14:02	<i>none</i>	   
BSL - Using the Stepper Using the Dr. Racket stepper to walk through the evaluation of an expression one step at a time.	5:36	stepper-starter.rkt	  
BSL - Discovering Primitives How to discover primitives beyond those covered in lecture. <i>Watch this lecture some time this week or next, but after lecture 1d.</i>	5:25	<i>none</i>	   

Practice Problems

These problems can help you practice the material in this module. We recommend that you do as many as you need to in order to become comfortable with the material before going on to the homework problems. Solutions are provided for practice problems, but **do not look at the solution until you have made a solid effort to solve the problem on your own**. Looking at the solution before trying to solve the problem will deprive you of most of the learning you would get from trying to solve the problem on your own.




Module Kind #	Assignment	Duration	Difficulty	Code Files	Requires Lecture
BSL P1	Write more arithmetic expressions.	5 min.		more-arithmetic-expression-starter.rkt more-arithmetic-expression-solution.rkt	BSL - exprs
BSL P2	Write out the step-by-step evaluation of expressions involving calls to primitives.	8 min.		evaluation-prims-starter.rkt evaluation-prims-solution.rkt	BSL - eval
BSL P3	Create an image using image composition primitives.	5 min.		tile-starter.rkt tile-solution.rkt	BSL - str-imgs
BSL P4	Write expressions to operate on strings using primitives.	8 min.		glue-starter.rkt glue-solution.rkt	BSL - str-imgs
BSL P5	Write expressions to operate on booleans using primitives.	7 min.		compare-images-starter.rkt compare-images-solution.rkt	BSL - bools-if-exprs
BSL P6	Step by step evaluation of a call to a function that calls a number primitive in its body.	7 min.		more-foo-evaluation-starter.rkt more-foo-evaluation-solution.rkt	BSL - stepper

BSL P7	Step by step evaluation of a call to a function that calls a string primitive in its body.	7 min.		even-more-foo-evaluation-starter.rkt even-more-foo-evaluation-solution.rkt	BSL - stepper
BSL P8	Write expression to produce background image of Canadian flag.	5 min.		cflag-starter.rkt cflag-solution.rkt	BSL - strs-imgs
BSL P9	Write even more arithmetic expressions.	5 min.		even-more-arithmetic-expression-starter.rkt even-more-arithmetic-expression-solution.rkt	BSL - exprs
BSL P10	Write an expression that operates on images using image primitives.	5 min.		triangle-starter.rkt triangle-solution.rkt	BSL - strs-imgs
BSL P11	Step by step evaluation of a call to a function that has an if expression in its body.	7 min.		bobble-evaluation-starter.rkt bobble-evaluation-solution.rkt	BSL - stepper
BSL P12	Fix the error(s) in a call to an image primitive.	5 min.		debug-rectangle-starter.rkt debug-rectangle-solution.rkt	BSL - strs-imgs

Homework Problems

You should complete these homework problems every module. You are encouraged to work with other students on these problems, but note two important rules: (1) Do not discuss the problem solutions on the forums; it is ok to use the forum to clarify the problem, but don't talk about how to solve it. (2) You must do the quiz entirely on your own.

Module				Code	Requires
--------	--	--	--	------	----------

Kind #	Assignment	Duration	Difficulty	Files	Lecture
BSL H1	Write an arithmetic expression.	5 min.		arithmetic-expression-starter.rkt arithmetic-expression-solution.rkt	BSL - exprs
BSL H2	Write an expression that operates on images using image primitives.	10 min.		overlay-starter.rkt overlay-solution.rkt	BSL - strs-imgs
BSL H3	Step by step evaluation of a call to a function that has an if expression in its body.	15 min.		foo-evaluation-starter.rkt foo-evaluation-solution.rkt	BSL - stepper

Module Quiz

Each module you will be asked to do an online quiz based directly on the material for the module. The quiz is designed to assess your understanding of the assigned homework problems — or more specifically of what you needed to understand to complete the homework problems. **You should complete the assigned homework problems before trying the quiz.** Please do not discuss quiz solutions on the discussion forums before they are due. You will have two attempts and a total of 24 hours to complete the quiz. Because the quiz is based on your mastery of the homework you should be sure to **complete the homework problems before** [starting on the quiz](#).

Tips for Success

General tips for doing well in this course can be found on the [To Do Well in This Course](#) page. The most important of these is that this is material that you **learn by doing**. Don't just watch the videos. You have to practice, practice, practice. So try typing along in Dr. Racket as you watch the video. Or, once the video is complete try to duplicate the work done in the video on your own. Definitely do the practice problems as well as the homework problems.

Also be sure to use your fellow students and the course staff as resources in helping you when you need it. Use the forums for asking questions.