

**CS 140: Computing Languages**  
University of Nevada, Las Vegas

## Instructor

Name: Guymon Hall  
Office: TBE-B382, Mo–Th 10:00am–12:00pm

Phone: 702-895-4852  
Email: [guymon.hall@unlv.edu](mailto:guymon.hall@unlv.edu)

## Course Objectives

After completing this course, you will be able to:

- Create programs in the Ruby programming language using sequential, selection, and repetition structures
- Utilize object-oriented programming techniques to develop robust programming solutions in the Ruby programming language
- Solve programming tasks by adapting Ruby programming language constructs to a specific solution

## Textbooks (Required)

*Learn to Program* (2nd edition), Chris Pine, The Pragmatic Bookshelf, 2013, ISBN-13 9781934356364

*The Well-Grounded Rubyist* (2nd edition), David A. Black, Manning Publications, 2014, ISBN-13 9781617291692

## Schedule of Topics

Ruby Basics		
Module 1	Introduction to Command-line Operations & Text Editors	Pine, ch. 1
Module 2	Programming Fundamentals & Installing Ruby	Pine, ch. 1
Module 3	Arithmetic, Assignment, Data Types, Variables, & Basic Input/Output	Pine, ch. 2–5
Selection, Repetition, & Collections		
Module 4	Relational Operators, Logical Expressions, & Conditional Statements	Pine, ch. 7
Module 5	Looping & Iteration	Pine, ch. 7
Module 6	Arrays & Hashes	Pine, ch. 7–8
Functional Decomposition & Methods		
Module 7	Methods, Parameter Passing, Return Values, & Code Blocks	Pine, ch. 9, 6
Mid-Term Exam		
Object-Oriented Programming		
Module 8	Abstract Data Types, Classes, & Objects	Black, ch. 2
Module 9	Class Methods & Variables	Black, ch. 3, 5
Module 10	Inheritance & Polymorphism	Black, ch. 3, 5
Beyond Ruby Basics		
Module 11	Modules & Multiple Inheritance	Black, ch. 4
Module 12	File Operations & Text Processing	Black, ch. 12
Module 13	Error Handling, Exceptions, & Rubygems	Black, ch. 6
Module 14	Rubygems & Beyond	
Final Exam		

## Academic Policies

### *Assignments*

There will be approximately 14 assignments worth 70% of overall grade.

### *Exams*

There will be a mid-term exam (worth 10%) and a final exam (worth 20%).

### *Grade Values*

Course grades will be assigned according to the following table (all values will be rounded to the next whole percentage point):

Grade	Percent
A	100%—90%
B	89%—80%
C	79%—70%
D	69%—60%
F	59%—0%

### *Grading Criteria*

Each homework assignment will indicate the number points available. All grades are subject to the discretion of the instructor, but in general grades will adhere to the following scheme:

- All assignments must be turned in on time. No late assignments will be accepted.
- All program submissions must be submitted via Webcampus in a single file named `rebelmailID.rb`. In the event of technical difficulties with Webcampus, assignments will be emailed to the instructor as an attachment.
- Programming assignments that violate academic integrity standards will be handled in accordance with the university's Student Academic Misconduct Policy, located at:

<http://studentconduct.unlv.edu/misconduct/policy.html>

- Program submissions that generate errors will receive a minimum deduction of 20 points. Program submissions that generate errors in an area that has been previously covered in the course will receive a minimum deduction of 50 points.
- **Coding style & documentation - 20%**
  - (5 pts) All programs must have the following comment block at the top of their main program file:

```
# Name: Your name, Class, Assignment number
# Partner: Your partner for this assignment (if any)
# Description: a brief description of the program
# Input: expected input to the program (if any)
# Output: expected output of the program (if any)
```
  - (5 pts) All programs must utilize proper coding style.
  - (5 pts) All methods and classes must have the following required documentation immediately above each method/class definition:

- # method\_identifier: brief description of what the method does.
  - # parameters: what to pass into the method
  - # return value: what the method returns, if any
  
  - # class\_identifier: brief description of the class
  - # public methods: a list of public methods identifiers
  - # instance variables: a list of all public instance variables
- (5 pts) All programs must have reasonable comments throughout the program that adequately convey what is going on the program (where it is not obvious).
- **Syntax & general knowledge - 30%**
    - (5 pts) All variables, methods, & classes must have meaningful identifiers.
    - (5 pts) All programs will refrain from using hard-coded values in the program; use named constants instead. Constant identifiers should be meaningfully named and should be in all caps with separate words using underscores.
    - (20 pts) Program submissions must contain zero syntax errors and must not generate any exceptions.
  - **Program solution - 50%**
    - (20 pts) Any required output from the program must be correct and must conform to the format specified in the assignment handout.
    - (30 pts) All programs must use all required techniques, data types, and data structures addressed in each assignment handout.

There will be no make-up exams except as covered under university policies.

### *Student Collaboration*

All students must submit their own work. Apart from formulating an assignment solution by one's self, acceptable sources of problem-solving input are limited to: instructor input, teaching assistant input, in-class examples, class web site examples, examples from the textbook, and official reference documentation.

Due to the nature of this course, collaboration among students enrolled in the class is allowed on a limited basis, provided such collaboration adheres to the stipulations in this policy. Student collaboration is defined as two or more students working together as a group to produce a programming solution. Student collaboration may occur for a given homework assignment under the following circumstances:

- Students enrolled in the class may work in pairs for each assignment on a mutually voluntary basis
- No more than two students enrolled in the class may work together for a given assignment
- Each student must turn in their own assignment
- Each student must indicate the name of their partner, if any, in the comment header section of each assignment
- Each student may work with the same partner for a maximum of three assignments throughout the semester

Violations of student collaboration may involve, but are not limited to: working with yet failing to identify a partner for an assignment, working with the same partner more than three times during the semester, asking other students to provide solutions for a programming assignment, copying code from other students, distributing code to other students, or copying, soliciting or distributing programming solutions to or from third-party sources (online forums or bulletin-boards, outside help from non-students, students from other classes, software repositories, etc.)

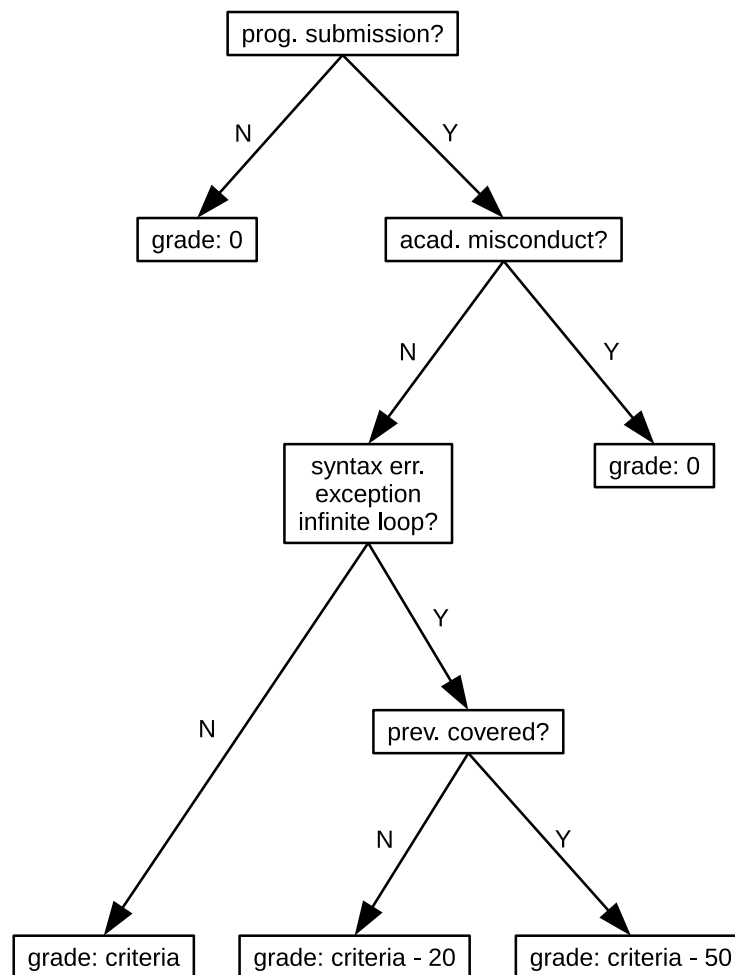
Note, this policy does not prohibit or discourage discussion among students involving the material covered in class. This policy is simply intended to discourage those who are not willing to make an honest effort at learning the various programming solutions we will encounter during the semester.

The grading policies for this course are designed to take into account the potential for unethical student collaboration should any violations occur. Thus, even if a student successfully violates this student collaboration policy, does not get caught, and gets perfect marks on his or her programming assignments, that student must still demonstrate his or her competency to a satisfactory level on the exam portions of the class. Each student must do his or her own work!

Should the need arise to deal with unethical student collaboration during the semester, the student(s) involved will be referred to the Office of Student Conduct under the [Student Academic Misconduct Policy](#). The general recommendation in this course for this policy is for the student to receive a 0 for that assignment for the first offense and a failing grade for the course for any subsequent offense.

### Grading Process

The figure below illustrates the grading process used to apply the listed criteria above in order to arrive at a grade for each homework submission:



In addition to the [Student Academic Misconduct Policy](#), the following university policies are applicable:

**Copyright**—<http://www.unlv.edu/provost/copyright>

**Disability Resource Center**—<http://drc.unlv.edu/>

**Religious Holidays Policy**—<http://catalog.unlv.edu/content.php?catoid=6&navoid=531>

**Transparency in Learning and Teaching-Higher Education (TILTHED)**

<https://www.unlv.edu/provost/teachingandlearning>

<https://www.unlv.edu/provost/transparency>

**Incomplete Grades**—The grade of I—Incomplete—can be granted when a student has satisfactorily completed three-fourths of course work for that semester/session but for reason(s) beyond the student's control, and acceptable to the instructor, cannot complete the last part of the course, and the instructor believes that the student can finish the course without repeating it. The incomplete work must be made up before the end of the following regular semester. If course requirements are not completed within the time indicated, a grade of F will be recorded and the GPA will be adjusted accordingly. Students who are fulfilling an Incomplete do not register for the course but make individual arrangements with the instructor who assigned the I grade.

**Tutoring and Coaching**—<http://www.unlv.edu/asc>

**UNLV Writing Center**—<http://writingcenter.unlv.edu/>

**Rebelmail**—By policy, faculty and staff should e-mail students' Rebelmail accounts only. Rebelmail is UNLV's official e-mail system for students. It is one of the primary ways students receive official university communication such as information about deadlines, major campus events, and announcements. All UNLV students receive a Rebelmail account after they have been admitted to the university. Students' e-mail prefixes are listed on class rosters. The suffix is always @unlv.nevada.edu. Emailing within WebCampus is acceptable.

**Final Examinations**—<http://www.unlv.edu/registrar/calendars>

## Course Communication

Students are encouraged to contact me at any time with questions regarding the class material. See the beginning of this syllabus for my contact information and office hours. (I am generally not available during weekends.) General information that might be useful by way of reference can be found on my departmental web page (<http://web.cs.unlv.edu/hallg2/>).

### *Discussion Forum Guidelines*

While there are a variety of publicly-available bulletin-board and forum web sites (Stack Overflow, StudyRoom, etc.), the primary location for such activity is restricted to the UNLV Webcampus Site. The Discussion Forum is there to encourage online discussion for general problem areas and questions over the material covered in class.

This presents a challenge in navigating the necessity of being able to freely discuss the course material while balancing the requirements of academic integrity standards. Thus, the following guidelines apply for online class postings in the Discussion Forum area:

- Posting code directly from assignments is prohibited, unless that code has already been distributed to the entire class
- Posting general solution steps is prohibited
- Asking general questions about how a certain aspect of a program works is allowed and encouraged!
- Answering such questions is allowed and encouraged!