CS 320 Programming Languages
Fall 2015

Credits: 3 units
Contact Hours: Monday and Wednesday 1400 – 1515
Instructors: Scott Lindeneau  Office: GMCS 562  Email: slindeneau@mail.sdsu.edu
Office Hours: Tuesday and Thursday 1200 – 1400 (and by appointment)

Course Materials
3. Course Reader, Aztec Publishing

Course Information for CS 320

Description from the Official Course Catalog
Principles of high-level programming languages, including formal techniques for syntax specification and implementation issues.
Prerequisites: Computer Science 108
Course Type: Required

Specific Goals for CS 320

Course-Level Student Learning Outcomes
1. Students will implement solutions to problems in a variety of programming languages.
2. Students will discover the strengths and weakness of various programming paradigms, including: imperative, procedural, functional, object oriented and logical programming.
3. Students will construct integrated solutions to problems utilizing several programming languages and paradigms simultaneously.
4. Students will assess problems, choose programming languages and/or paradigms and then implement solutions.
5. Students will explain and use common language concepts such as; scope, types, binding, lexical analysis, syntax analysis, control constructs, functions, data management, error handling and concurrency.
6. Students will discuss various implementation methods and design decisions for/within programming languages.
7. Students will use multiple forms of OOP in various programming languages.

Relationship to CS Program Course Outcomes
CS 320 addresses the following CS Program course outcomes:

a) An ability to apply knowledge of computing and mathematics
b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
d) An ability to communicate effectively with a range of audiences
e) Recognition of the need for and an ability to engage in continuing professional development
f) An ability to use current techniques, skills, and tools necessary for computing practice
g) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.

h) An ability to apply design and development principles in the construction of software systems of varying complexity.

**Topics Covered**

The following topics are covered in CS 320:

1. Programming Language History.
2. Syntax, Semantics, Parsing and Lexical Analysis.
3. Prototype and Class based OOP.
4. Early/Late Binding, Scope and Data Types.
5. Abstract Types, inheritance, polymorphism.
6. ControlConstructs & Function Invocation.
7. Imperative, Procedural, OOP, Functional and Logical Programming
8. Language Abstractions, Cohesion/Coupling and Program Design.
9. Data types, Typing and Typing Conventions.
10. Expressions, Statements and Control Structures
11. Functions (Sub-Programs) and Recursion.
12. Concurrency.

**Course Schedule and Grading Policies**

**Approximate Due Dates for Major Assignments and Exams**

<table>
<thead>
<tr>
<th>Weeks</th>
<th>Topic</th>
<th>Project Activities/ Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction, Libraries Language: Intro C</td>
<td>Homework 1: Due Week 2. Reading: Ch 1</td>
</tr>
<tr>
<td>2</td>
<td>Binaries &amp; Libraries, Command Line, Conceptual Abstraction, Shell Intro Language: Shell scripting</td>
<td>Homework 2: Due Week 3. Reading: Ch 2</td>
</tr>
<tr>
<td>3</td>
<td>Syntax and Semantics Language: C</td>
<td>Homework 3: Due Week 4. Reading: Ch 3</td>
</tr>
<tr>
<td>4</td>
<td>Parsing/Lexical Analysis, Binding, Scope Language: Shell scripting &amp; C</td>
<td>Homework 4: Due Week 5. Reading: Ch 4</td>
</tr>
<tr>
<td>5</td>
<td>OOP, Bindings &amp; Scope Language: Shell scripting &amp; C++</td>
<td>Homework 5: Due Week 7. Reading: Ch 5</td>
</tr>
<tr>
<td>6</td>
<td>OOP Syntax &amp; Data Types Language: Shell scripting &amp; C++</td>
<td>Reading: Ch 6</td>
</tr>
<tr>
<td>7</td>
<td>Scope, Data Types &amp; First Order Data Language: Lua</td>
<td>Homework 6: Due Week 8. Reading: Ch 9</td>
</tr>
<tr>
<td>8</td>
<td>Language Integration / Extensibility Language: Shell, C/C++, Lua</td>
<td>Homework 7: Due Week 10.</td>
</tr>
<tr>
<td>9</td>
<td>Glue Code Language: C/C++, Lua</td>
<td>Midterm #1 Reading: Ch 10</td>
</tr>
<tr>
<td>Weeks</td>
<td>Topic</td>
<td>Project Activities/ Reviews</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>----------------------------</td>
</tr>
</tbody>
</table>
| 10    | Expressions  
Language: Python | Homework 8: Due Week 12.  
Reading: Ch 7 |
| 11    | Abstract Data Types  
Language: C/C++ & Python | Reading: Ch 11 |
| 12    | Prototyped Objects & OOP  
Language: C++, Lua & Python | Homework 9: Due Week 14.  
Reading: Ch 12 |
| 13    | Prototyped Objects & OOP  
Language: C++, Lua & Python |  |
| 14    | Functional Programming  
Language: Haskell | Homework 10: Due Week 16.  
Reading: Ch 15 |
| 15    | Logic Programming  
Language: Prolog | Reading: Ch 16 |

**Evaluations**

1. Ten Assignments (50% of final grade, 5% each)
   1) First C Program. Evaluates LO 1,2
   2) Shell scripting. Evaluates LO 1,2
   3) C programming and the global scope. Evaluates LO 1,2
   4) Programs as Functions & Data management. Evaluates LO 1,2,3
   5) C++ OOP. Evaluates LO 1,2,3,4,5
   6) Lua Scope. Evaluates LO 1,2,3,4,5
   7) Lua Extensions. Evaluates LO 1,2,3,4,5
   8) Python LO 1,2,3,4,5,6,7
   9) Prototyped-based vs Class-based Objects LO 1,2,3,4,5,6,7
  10) Haskell & Functional Programming LO 1,2,4,5,6

2. Tests (50% of final grade)
   1) One midterm. (20% of final grade) Evaluates LO 1,2,3,4,5,6
   2) Final. (30% of final grade) Evaluates LO 1,2,3,4,5,6,7

**Grading Scale:**

- A  93% and above
- A–  90%
- B+  87%
- B  83%
- B–  80%
- C+  75%
- C  70%
- D  60%
- F  below 60%
Other Course Policies

Special Assistance: If you are a student with a disability and believe you will need accommodations for this class, it is your responsibility to contact Student Disability Services at (619) 594-6473. To avoid any delay in the receipt of your accommodations, you should contact Student Disability Services as soon as possible. Please note that accommodations are not retroactive, and that accommodations based upon disability cannot be provided until you have presented your instructor with an accommodation letter from Student Disability Services. Your cooperation is appreciated.

Cheating Policy: There is a zero tolerance policy on cheating in this course. You are expected to complete all programming assignments on your own. Collaboration with other students in the course is not permitted. You may discuss ideas or solutions in general terms with other students, but you must not exchange code. (Remember that you can get help from me. This is not cheating, but is in fact encouraged.) I will examine your code carefully. Anyone caught cheating on a programming assignment or on an exam will receive an "F" in the course, and a referral to Judicial Procedures.