

CS331 – Programming Languages

Syllabus – Spring 2014

INSTRUCTOR: Tom Naps

TEACHING ASSISTANT: Sam Micka

OFFICE: Halsey 214, phone 424-1388

EMAIL: naps@uwosh.edu

OFFICE HOURS: Tues & Fri: 9:00 – noon. And by appointment.

REFERENCES AND RESOURCES:

- Slides posted on D2L by 8:00am the day of a class session. Pick them up and bring them to class so you can take notes on and about them. They are not meant to be a “book” but rather to provide high-level organization to what we do in class. If they are not liberally saturated with your own explanatory notes, they will likely prove useless when you need them most – for example when you work on an assignment, do the daily review problems, or prepare for an exam.
- Review problems posted on D2L at the end of each class. Complete them by noon on the day of the following class meeting and be ready to discuss the answers you have submitted (e.g., review problems posted Wednesday are due by noon Friday).

Topic Coverage

1. The functional programming paradigm
 - In a functional language with dynamic scoping and weak typing
 - In a functional language with static scoping and strong typing
 - In the formal λ -calculus
2. Formal Syntax
3. Writing an Interpreter for a small language – Is it a toy or isn't it?
4. Evaluation Order and Parameter-passing
5. Memory management
6. The logic programming paradigm
7. Parallel Programming

Learning Outcomes

1. Given an English description of a formal language, the student will be able to construct a context-free grammar that generates this language.
2. Given a context-free grammar and the source code for a program, the student will be able to parse the program according to the grammar, and to produce a parse tree of the program or identify syntactic errors in the program.
3. Given a context-free grammar in Backus-Naur Form (BNF), the student will be able to convert it to an equivalent, more compact grammar in Extended BNF.
4. Given a context-free grammar, the student will be able to determine whether the grammar is ambiguous or not.
5. Given a program and a scoping mechanism (static or dynamic), the student will be able to trace the execution and infer the output of the program.
6. Given a formal description of an operation, the student will be able to implement it in the functional paradigm using either recursion or a computational pattern such as filtering, mapping, or folding, or a combination thereof.
7. Given an imperative program and a set of eager/lazy parameter-passing mechanisms, the student will be able to simulate, for each mechanism, the sequence of updates that take place in memory as the program executes.
8. Given the description of an operation applicable to an infinite data structure, the student will be able to program this operation in a functional language using lazy evaluation.
9. Given a functional language with higher-order functions, the student will be able to simulate recursion using the Y combinator.
10. Given a working interpreter for a programming language, the student will be able to adapt the interpreter to a similar language with a different concrete syntax.
11. Given a working interpreter for an imperative programming language, the student will be able to implement an interpreter for an enhanced language with additional features (such as a new data type or a new language construct), or different semantics (such as a different parameter-passing mechanism).
12. Given a problem specification, the student will be able to formulate a solution to it in a logic programming language such as Prolog.
13. Given a problem specification, the student will be able to program a parallel algorithm to solve the problem more efficiently than a sequential program would.

Course Grading Policies

Your grade for the course will be based on the following weighted factors:

Factor	Weight
6-8 Assignments	40% in total
Class participation and preparation	16%
Exams:	44%

To get the 44% contribution to your grade from the four exams, I will use the formula:

$$E = 0.055 \times E_{worst} + 0.165 \times E_{best} + 0.11 \times E_{other1} + 0.11 \times E_{other2}$$

where E_{worst} is your worst exam score and E_{best} is your best exam score.

At the end of the term, your work in all of these areas will contribute to a numerical grade for the course based on a 100-point scale. Grade cutoff levels on this final scale are:

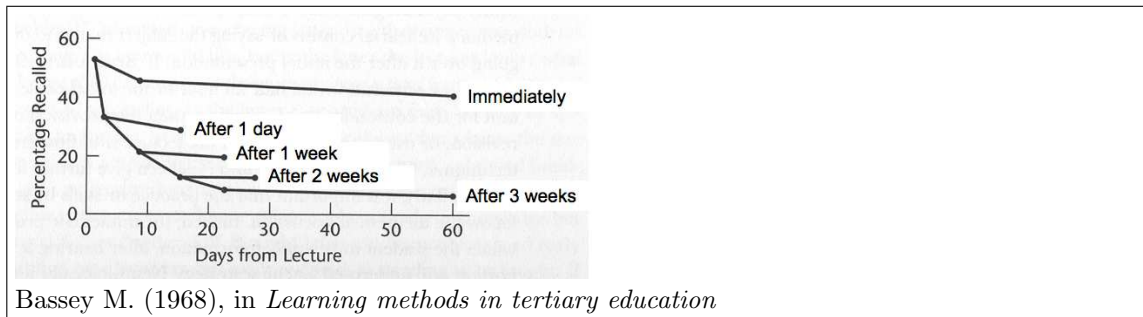
A \geq 92	B \geq 82	C \geq 72	D \geq 62
A- \geq 90	B- \geq 80	C- \geq 70	D- \geq 60
B+ \geq 88	C+ \geq 78	D+ \geq 68	F < 60

FAQ

Do I have to come to class? You are expected to arrive prepared to ALL the course sessions. Furthermore you are expected to participate in the classroom discussions and activities to the best of your abilities. This includes being ready to defend your answer to the review problems from the previous class (more on that later). It is difficult to envision a student missing and/or arriving unprepared to a number of the class sessions and still succeeding in the course.

How much time will this course take? Figure about three hours outside of class for each hour in class. That heuristic makes being a full-time student pretty much equivalent to holding a full-time job, so this is really good preparation for the real world that awaits you after graduation.

How can I best prepare for the exams? We've known what the following graph illustrates since 1968:



More interesting evidence ...

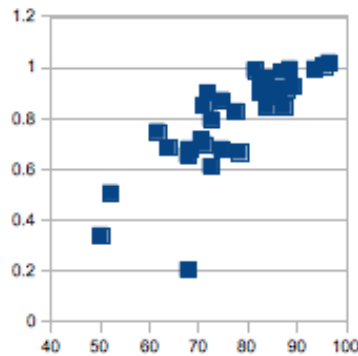
“Research has demonstrated that after a lecture, students recall 62% of the information. However, only 45% is recalled by students after 3-4 days and in 8 weeks only 24% of the information is recalled. If a quiz or exam was administered after the lecture, recall was doubled at the 8-week period. It is interesting that many faculty members appear to ignore the potential impact which quizzes and tests can have upon learning.” – Bonwell C.C., Eison J.A.: *Active Learning: Creating Excitement in the Classroom*. Washington, DC: George Washington University, 1991.

Consequently at the end of most of our class periods you will find on D2L a small set of review problems covering what we discussed in class. The time to work on these review problems is immediately after the material is covered in class. You are also encouraged to discuss review problems with your classmates in a spirit of mutual help toward better understanding of how to solve them.

We will always discuss the review problems at the beginning of the class following their distribution. Your solutions to the review problems are due no later than noon on the Monday, Wednesday, or Friday following the day of their distribution. These solutions should be submitted using the link to a blank “quiz” for the review problems that you will find on D2L.

If you have participated in class the day the review problem was distributed, have made a good faith effort to work on the review problem, and are “stuck” on it, I will be more than happy to help you with it if you come my office anytime within

three days after you have received the review problem in class. After those three days (not counting weekends), *because you have made the choice to not learn effectively*, you are on your own in terms of grappling with these review problems. Although the review problems only count 16% of your grade, the following correlation from a previous course between review-problem-percentage (on a 0 to 1.0 vertical scale) and overall percent in the course (on a horizontal scale of 0 to 100) is indicative of their true importance.



What if I'm late in submitting an assignment for evaluation? Each assignment will carry with it a due date. If you are late in submitting it for evaluation, it will be accepted but will be penalized at the rate of 10% of point value the first day late, *an additional 20%* the second, *an additional 30%* the third ...

Is there any way I can carelessly lose points in the course? Yes ...

- Be late in submitting your work for evaluation on assignments.
- Don't participate in and prepare for the class.

What is this class participation/preparation stuff? How does it add up to 16% of my grade? ...

- Be sure to get those review problems done and submitted on time
- Exhibit your knowledge when called on to explain your answer to a review problem.
- Exhibit your knowledge when called on to respond to other questions in class