CISP300(13884)

Tak Auyeung, Ph.D.

January 21, 2011

1 Course Information

- *Title:* Algorithm Design/Problem Solving
- *Description:* This course introduces methods for solving typical computer problems through algorithm design. Topics include assessing and analyzing computer problems in a top-down, divide-and-conquer approach that leads to a programming solution. It also covers programming plans and detailed design documents from which source code versions of programs are created.
- *Student Learning Objectives:*
  - define operators, including arithmetic, comparison, and logical operators.
  - differentiate control structures, including branches (conditional statements) and loops (pre-checking and post-checking loops).
  - deduce post condition from pre condition for control structures, including assignment statements, branches, and loops.
  - construct a trace table to emulate the execution of a program that utilizes variables, various control structures, data organizations, subroutines, and parameters.
  - contrast the lifespan limits and behaviors of local variables, by-value parameters, and by-reference parameters.
  - compare the two methods of passing results: by-reference parameters and return value.
  - compare in-line copy-and-paste coding with structured subroutines in terms of maintainability, defect containment, testability, and other metrics.
  - synthesize a subroutine to abstract one or more similar blocks of in-line code using local variables, parameters, and return values.
  - differentiate roles involved in software development, including developers, analysts, and test engineers.
- *Number of units:* 3.00 lecture
- *Meeting hours:* 54 lecture hours

2 About the Instructor

Name	Tak Auyeung
Office	#7 in Liberal Arts #133
Office Hours	M-Th: 0930-1030, F: 1105-1205 (main campus)
Phone	916-484-8250
Email	tauyeung@drtak.org or auyeunt@arc.losrios.edu
Website	www.drtak.org/teaches/ARC
Online Classroom	https://www.someprofs.org/moodle

3 Section information

- *Meetings:* F: 0800-1105
- *Final exam:* 2011/05/06 0800-1000

4 Class Policies and Rules

All policies of this class are extensions to the ones already imposed by the Los Rios Community College District and American River College.

4.1 Acceptable Excuses

- Sickness of self with doctor's note, dated and signed.
- Jury duty with a copy of the summon.
- Military duty with a copy of the notice.

4.2 Attendance (quoted from the college catalog)

College students are expected to attend all sessions of their courses. Excessive absence may result in the student being dropped from class by the instructor. *A student may be dropped from any class when that students absences exceed six percent (6%) of the total hours of class time.*

If a student is absent because of illness verified by the Health Center or personal physician, the absence must be excused and the student allowed to make up work missed.

My suggestion: if you are sick enough and/or contagious enough to miss a class, get some form of verification by the campus Health Center or personal physician. Non-verified sickness related absences are not excused.

Note that I am not required to give any warning before removing a student from a class with excessive absences.

4.2.1 Attendance for Online Classes

Attendance for an online class is determined by online access frequency. Absence is defined as the lack of observable activity. For example, in a 16-week full semester class, the duration is $16 \times 7 \times 24 = 2688$ hours. 6% of that is 161 hours, or roughly 6 days. As a result, an interval of at least 6 days between activities is considered excessive absence.

4.2.2 Attendance for Face-to-face/Hybrid Classes

Attendance for a face-to-face or hybrid class that has mandatory face-to-face time is determined by roll taking by a variety of means. Roll can be taken by a roll sheet, an in-class quiz, an exam or other means.

4.3 Classroom and Lab Behavior

- The classroom and lab are facilities provided only for the intended course work.
- Any behavior in the classroom or lab that interferes with teaching or learning is not tolerated. This includes disruption, bullying, excessive chatting and etc. The following is a list of examples.
  - Chatting. If there are any class related questions, ask me instead of another student.
  - Pets. Unless it is a legitimate helper dog, please do not bring any pets to a class.
  - Cell phones ringing, especially those with "interesting" ring tones.
- No eating or drinking is permitted in the classroom or lab. Food and drink items are to be sealed or closed during a lecture or lab session.
- Non-compliant participants will be asked to leave the classroom or lab. Further non-compliance will result in the involvement of security personnel, and will be reported to the Student Discipline Officer. This can result in suspension or expulsion.

4.4 Conduct code

- Refer to the section "Student conduct code" of the college catalog for a full description. You are expected to read and understand that section prior to the beginning of the second class meeting for a face-to-face class, or 3 days after the beginning for an online class.
- Academic dishonesty is an act of deception in which the student claims credit for work or effort of another person or uses unauthorized material or fabricated information in any academic work. It occurs when students (participate in any activity to) attempt to show possession of a level of competency, knowledge or skill that they do not possess.
- Any academic dishonesty may be reported to administration. An accumulation of reported attempts, possibly from multiple classes, will lead to expulsion from the college.
- All participants involved in academic dishonesty do not receive points for the worked submitted, this includes the person or people who originate the material, but willingly, actively or otherwise knowingly involve in an attempt.
- Activities involving academic dishonesty do not count for attendance.
- The instructor may ask suspects to do additional work or explain submitted work in order to determine the legitimacy of submitted work.
- By default, all work (assignments, quizzes, exams) are expected to be completed as follows:
  - Independently: this means without the aide (specific to the work) of anyone during or prior to the activity. This means that acquiring questions or test banks prior to an exam or quiz is considered an act of academic dishonesty.
  - Originally: this means without copying from or significantly deriving from the work of others.
- I can retroactively reexamine submitted *and* graded work for evidence of academic dishonesty. As a result, I can also adjust the grade accordingly.

5 Grading

5.1 Letter Grade Equivalence

- F: $x < 1/8 = 12.5\%$
- D: $1/8 = 12.5\% \leq x < 3/8 = 37.5\%$
- C: $3/8 = 37.5\% \leq x < 5/8 = 62.5\%$
- B: $5/8 = 62.5\% \leq x < 7/8 = 87.5\%$
- A: $7/8 = 87.5\% \leq x$

5.2 Components and Proportions

- Homework or in-class assignments (20%).
  - Mostly homework.
  - Mostly without any time limit.
  - A chance to practice learned material.
  - Answers disclosed when it is due.
- Final exam (40%):
  - Proctored or in person.
  - Time limit of 2 hours.
  - Comprehensive.
  - As scheduled in the class schedule.

- Exam 1 or Project 1 (20%)
  - Occurs at about 1/3 through the class.
  - First formal assessment, proctored if it is an exam.‘
- Exam 2 or Project 2 (20%)
  - Occurs at about 2/3 through the class.
  - Second formal assessment, proctored if it is an exam.

5.3 Grading FAQs

- Score is assigned based on *observable academic competence*, not effort.
- Multiple choice questions are balanced, this means the total score of correct answers equal the total score of incorrect answers of the same question.
- Some questions are graded based on “key indicators”, which are objective observable characteristics of the submitted work.
- This is no late submission. Submissions after due date/time do not count or are not permitted, unless covered by an excuse (see the accepted excuses section).

6 Tips

These are not rules for this class, just tips from your instructor.

6.1 Helping a fellow student

I don't have any problem with students helping out each other. Frequently, the helper and helpee both benefit. However, helping is not letting a friend copy answers, or completing assignments/test questions for someone else.

When does helping a friend become academic dishonesty? Let's say Pat helped Sam with an assignment. If the instructor determines that Sam's submission shows a competency that Sam does not have, then it is considered a case of academic dishonesty.

But isn't this a gray line? Not exactly. Sam's true competency can be re-assessed by a similar but different assignment, completed under supervision. If Sam's result of the two assignments are significantly different, then the first submission shows a competency that Sam does not have.

What does this mean when you want to help a friend? Make sure the helpee truly understands the concepts and develops the necessary skills. Make sure your helpee can get a similar score if the instructor is to reassess the helpee's competency using a different assignment.

6.2 Financial aide

Students who are on financial aide should consider that aspect carefully. I have had students who fail a class and pleaded for financial reasons. Unfortunately, the grade of a class is based on academic performance, I cannot adjust it based on financial needs.

Sadly, there were also cases where students cheated in classes, and ended up failing. This is worse than just failing a class because reported cases of cheating can leave a record in the office of the college Discipline Officer.

6.3 Enrollment of future classes

There have been cases where a student thought he/she would pass a class, and ended up failing. As a result, the said student could no longer meet the prerequisite of a class that he/she had enrolled in the following semester.

This happens mostly because a student cheated, and I retroactively regraded the work(s) that involved dishonesty. In other words, the mid semester grade indicated that the said student would pass a class with a letter grade of C or better. However, due to the retroactive regrading, the actual final grade was D or lower.

6.4 Study habit

First, attend classes. Many of my classes require the attendance of every class to keep up with the topics. Missing one class can make subsequent class difficult to follow. Furthermore, missing 6% or more class meetings (unexcused) will lead to being dropped.

Second, pay attention in class. Bring *something* to take notes. Just because I may record my lectures doesn't mean there is nothing to do. There may be something that you realize that I do not actually say, write that down!

Reading what I will be covering before a lecture will help you prepare for a lecture. Even if you don't fully understand what you read, at least you can prepare questions to ask me during the lecture. A pre-exposure to concepts and topics can help you prime your mind for discussions in a lecture.

Study again after a lecture. Revisit tricky topics, review notes, repeat examples, etc. Do this as soon as possible so that memory of the lecture does not fade away.

Start on assignments as soon as possible. Starting early does not create additional work. You have more time to ask questions, and more buffer in case anything happens before the due date.

After an assignment is due, study my solution. You may get full points for an assignment, but you may still learn something important from my solution. Do this as soon as possible so that you can use my solution to better understand course topics.

I use a lot of examples in a class that are not in my notes. Review and study them. Anything that I present in a lecture is in the scope of assessments.

7 Schedule

Topic	begin date
The different roles involved in software development: systems analysts, developers and test engineers. Necessary training for programmers. Special job requirements for programmers. The definition of "algorithm". How algorithms relate to programmers.	01/21/11
The definition of "variable". Assignment statements as a means to change the values of variables. How to interpret assignment statements that have the same variable on both sides of the operator. Sequences of assignment statements. The technique to trace a sequence of assignment statements.	01/21/11
Conditional statements using diagrams. The behavior of conditional statements. How to trace the execution of a conditional statement.	01/28/11
Pre-checking loops using diagrams. Post-checking loops using diagrams. How to trace the execution of loop statements.	02/04/11
The importance of structured programming. The importance of indentation. The behavior of a nested statement involving conditional statements and loops. How to trace the execution of a nested statement.	02/18/11
Pre and post conditions. How to compute the post condition of an assignment statement that is not self-referencing. How to compute the post condition of an assignment statement that is self-referencing. Post conditions and algorithm validation.	02/25/11
How to compute the post condition of a conditional statement. How to compute the post condition of a loop statement. Loop invariant.	03/04/11
The necessity of top-down design. Top-down design using a complex algorithm.	03/11/11
The limitations of algorithms without arrays. The definition of "array". The definition of the indexing operator. Simple algorithms using arrays.	03/11/11
Short circuited Boolean expression evaluation. Why short circuited Boolean expression evaluation is important. How to emulate short circuited Boolean expression evaluation when it is not implemented by a programming language.	03/18/11
How to search in an array using a linear search algorithm. How to search in an array using binary search.	03/25/11
Subroutines. How to trace the execution of subroutines. How to trace recursive subroutines.	04/01/11
The limitations of using global variables. Local variables. The lifespan of a local variable. The importance of initializing a local variable. How to represent a local variable in a trace.	04/01/11

The main role of parameters. By-value parameters. How a by-value parameter relates to a local variable. By-reference parameters. How to represent parameters in a trace.	04/08/11
An alternative method to pass a single result from a subroutine. Return statement. The behavior of a return statement. How to trace the execution of a subroutine that returns a value.	04/08/11
The problems of copy-paste-and-modify. How to reuse the same code at difference places in an algorithm using subroutines.	04/22/11
The importance of abstracting a segment of code into a subroutine. How to classify variables used in an algorithm into local variables, by-value parameters and by-reference parameters. How to track local variables and parameters in a trace.	04/29/11
The necessity of types. How a type restricts the possible operations on a value. How to designate the types of variables and parameters.	04/29/11
Record (structure) as a mechanism to bundle related variables. How to access various parts of a record. How a record can contain arrays. How an array can contains records.	05/06/11
File concepts. File open, read, write, and check for end-of-file operations. Algorithms using file concepts. How to trace algorithms using file operations.	05/13/11