DE ANZA COLLEGE **Spring, 2015**

43675 CIS -022C-61Y ***Data Abstraction and Structures*** *(4 1/2 units)*

Syllabus

Instructor: Shu-huar Yeh

Office Location	**:** n/a
Phone	**:** n/a
Email	**:** yehshuhuar@fhda.edu
WEB	**:** http://puma.deanza.fhda.edu/distribute/yehsh/cis-22c/
Hours	**:** n/a

Class meetings:	AT312	Tue, Thu	8:00PM – 9:50PM
LAB	**:** ONLINE	See calendar entries colored in green	

Prerequisites: Computer Information Systems 22B or equivalent. Advisory: Mathematics 212 or equivalent.

Course Description: Application of software engineering techniques to the design and development of large programs; data abstraction and structures and associated algorithms: stacks, queues, linked lists, trees, graphs, and hash tables; internal and external sorting; use of recursion; team project.

Text required: *Data Abstraction and Problem Solving with C++, 6th edition*
ISBN: 978-0132923729 by Carrano

Academic support: The Tutorial Center provides free academic support and free workshops open to all De Anza students, and Adjunct Skills and Self-Paced
Skills courses: http://deanza.edu/studentsuccess/

Student Learning Outcomes: By the end of the course, students will:

- Design programs using abstract data type concepts
- Write programs using stacks, queues, linked lists, heaps, trees and graphs
- Implement recursive algorithms
- Implement hashing algorithms
- Implement advanced sorting: shell, heap, quick, and external sorts

Attendance policy: This 4.5 Units course consists of 4 lecture hours (4x50min = 200min) and 1.5 lab hours (75min) per week. You are expected to attend all lecture and lab sessions. If you must be absent from class, arrange with another student to share class notes for that session. You should plan on spending approximately 12 to 15 hours per week to study and do your homework. If you wish to drop the class, it is your responsibility to do so. An unauthorized withdrawal from class without following official procedures will result in you being assigned a grade of "F" (or "NC" if you have selected the Credit /No Credit option).

Important Dates on Add and Drop
http://www.deanza.edu/calendar/springdates.html

Scholarly conduct: In order to be successful in this class you will have to make a commitment to studying, reading the text, doing your homework, writing your lab assignments, attending class, and taking notes. Worthwhile contribution and regular attendance can positively affect the grades. You are expected to do your own work. Cheating or plagiarism in any form will not be tolerated. Copying or cheating during a test will result in a zero being assigned for that test and may result in a failing grade for the entire course. Any copied assignments will result in a zero grade for all parties, and may result in a failing grade for the entire course. It may also result in dismissal from class, college disciplinary action, and/or notation in their permanent records. The Business Division Dean will also be notified by letter. Please check the current Schedule of Classes to learn more about academic integrity, other policies, and Student Standards of Conduct (http://www.deanza.edu/studenthandbook/academic-integrity.html).

Homework: Homework is not to be run on the computer (unless you wish to). The purpose of the homework is to help clarify the material for you as we proceed and to prepare you for tests. You don’t need to turn in the homework exercises, but you are strongly encouraged to do it.

Class and online assignments: There will be given between 10 to 20 assignments to be completed in class or on line, such as group work, quizzes, etc. They are open book, open notes. There is no make up for any such assignment. These are pass/no pass type of assignments.

Tests: There will be pop quizzes throughout the course. There will be no make-up for the quizzes you have missed. There will be one midterm exam (100 points) and a comprehensive final (120min, 100 points). Both tests are open book, open notes. Test dates are shown on the calendar. Final exams papers will be retained for a period of 90 days from the exam date. The final exam will be similar to the midterm exam, emphasizing the material learned after the midterm but covering the rest of the quarter as well.

Laboratory assignments: You will be given 5 programming assignments (100 points each) and a team project (100 points). They are to be run using the computer.

- Each assignment must be submitted on or before the due date.
- No assignment will be accepted after the due date.
- Partial credit will be given for incomplete assignments.
- There is no make up for any assignment.
- All assignments must be copied to the specified file system on campus including all source and header files and any output generated by the program.
- If the first three programming assignments are satisfactory you will be given a team project (2-5 students in a team).
- For more information on the grading of lab assignments, see “How Lab Works Are Evaluated”

Extra-credit may occasionally be given throughout the course (up to a total of 9 points).

Grading: To pass the class you have to do the following:

1. classAssignmentsCompleted >= 80%

2. labAverage (average of the 5 programming assignments and the team project considered twice) >= 70
3. midTerm >= 60
4. final >= 60

If all of the above are true, your final grade will be calculated as follows:

score = (labAverage + midTerm + final + extra credit) / 3

Your grade is based on the score you earn as shown below. Worthwhile contribution and regular attendance can positively affect the grades.

Percentage	Total Points	Grade
90 – 103		A-, A, A+
80 – 89.9		B-, B, B+
70 – 79.9		C, C+
60 – 69.9		D
0 – 59.9		F

Tentative Calendar

	Topics	Sun	Mon	Tuesday	Wed	Thursday	Fri	Saturday
April 1st week	ADT, Lists Chaps. 1, 3, 4, 8, 9	4/5	6	7	8	9 online lab 6 -7:15 pm	10	11
2nd week	Stacks 6, 7	12	13	14	15	16 online lab 6 -7:15 pm HW1 Due	17	18
3rd week	Queues 12, 13, 14	19	20	21	22	23 online lab 6 -7:15 pm HW2 Due	24	25
4th week	Recursion 2, 5, 10	26	27	28	29	30 online lab 6 -7:15 pm	5/1	2
May 5th week	Trees 15, 16	3	4	5 Read and understand team project requirements. Create teams	6	7 online lab 6 -7:15 pm HW3 Due	8	9
6th week	Heaps 17	10	11	12 Team Project: Report 1 – due	13	14 online lab 6 -7:15 pm MIDTERM	15	16
7th week	Dictionaries, Balanced Search Trees 18, 19	17	18	19 Team Project: Report 2 – due	20	21 online lab 6 -7:15 pm HW4 Due	22	23
8th week	Graphs 20	24	25	26	27	28 online lab 6 -7:15 pm	29	30
June 9th week	Sorting 10, 11	31	6/1	2 Team Project: Report 3 – due	3	4 online lab 6 -7:15 pm HW5 Due	5	6
10th week	External Storage 21	7	8	9 Team Project: Report 4 – due	10	11 online lab 6 -7:15 pm	12	13
11th week		14	15	16 Project	17	18 Project	19	20
12th week	Final Exam Week	21	22	23 Final Exam 8:00 pm – 10:00 pm	24	25 NO CLASS	26	27

Here is the login information for the online labs. (Normally you will hear my voice from your computer speaker and the phone is not needed.)

CCC Confer

Meet & Confer

RECURRING **MEET & CONFER PARTICIPANT INVITATION**

Meeting Details

Status: Active
Title: RECURRING CIS 22C Online Lab
College/Group/Org: De Anza College
Closed Caption: No

Date	Start time	End time	Duration	Subtitle	Closed Caption
4/7/2015	8:00 PM	10:00 PM	120	Test Session	No
4/9/2015	6:00 PM	7:30 PM	90	Spring 2015	No
4/16/2015	6:00 PM	7:30 PM	90	Spring 2015	No
4/23/2015	6:00 PM	7:30 PM	90	Spring 2015	No
4/30/2015	6:00 PM	7:30 PM	90	Spring 2015	No
5/7/2015	6:00 PM	7:30 PM	90	Spring 2015	No
5/14/2015	6:00 PM	7:30 PM	90	Spring 2015	No
5/21/2015	6:00 PM	7:30 PM	90	Spring 2015	No
5/28/2015	6:00 PM	7:30 PM	90	Spring 2015	No
6/4/2015	6:00 PM	7:30 PM	90	Spring 2015	No
6/11/2015	6:00 PM	7:30 PM	90	Spring 2015	No

Is Your Computer Ready?
How to Connect with Your Mobile Device

Participant Details
Telephone conference line: 1-913-312-3202 *
Participant passcode: 677590
*Toll free number: 1-888-886-3951

Go to www.cccconfer.org
Click Meet & Confer Participant Log In
Locate your meeting and click Go
Passcode: 677590

Telephone Conference Feature
*6 - Mute/unmute your individual line

FOR ASSISTANCE
CCC Confer Client Services - Monday - Friday between 8:00 am - 4:00 pm
Phone: 1-760-744-1150 ext 1537 or 1554
Email: clientservices@cccconfer.org

How Lab Works Are Evaluated?

An individual lab initially will receive a minus (-), a checkmark (√), or a plus (+) based on the following criteria:

1. How well is the program structured? For example, are there divisions into smaller modules using functions and classes?
2. How well is the program documented? For example, have you described the purpose of each module and the expected result of each function or method?
3. How well is the program written? Have you written it according to the programming styles described in Appendix B of the textbook? Have you used proper naming conventions for the identifiers? Is the program easy to read?
4. How well does the program satisfy the requirements?
5. How does the program execute?

Based on the assessments above, a score from 0 to 100 will be assigned to all the labs submitted by the student through ranking and by curve.

How Do You Submit Your Lab Assignments?

1. You need to copy your files to the file system location (usually on the G: network drive) specified by the instructor using one of the lab computers.
2. The following files are required:
   a. The program source files with ".cpp" extension using the naming convention, "**Labxy_firstName_lastName.cpp**". For example, Lab1a_Jane_Doe.cpp, Lab1b_Jane_Doe.cpp, etc. Each program should start with a multiline comment that includes the student's name, detailed description of the program, the date the program is written, and the IDE used to create the program.
   b. Each program should have a text file showing the result of a sample run. That includes all keyboard input and screen output. The text file for a program should be named using the program file name with the .txt extension. For example, Lab1a_Jane_Doe.txt.
   c. Any input files required by your programs.
   d. Any output files created by your programs.

Check list before submission of your labs

Files	Source files, input/output files, interaction files, diagrams (*)
Program description	Multiline comments at the beginning of your code including student name, date, description, etc.
Programming style	Alignment of parts and indentations.
Documentation	Relevant comments, the more the better.
Program build time	Neither warning messages nor error messages are accepted.
Program run time	Program should run from start to finish without any errors.

(*) UML sequence and class diagrams are optional. You may gain extra credit for drawing those diagrams. The diagrams should be drawn using a tool such as PowerPoint. Drawing by hand on a piece of paper and scanning it into an image is not acceptable.

Program Template

Use the following template for your labs. Each lab should contain one single source cpp file.

File Name: Lab1a_firstName_lastName.cpp

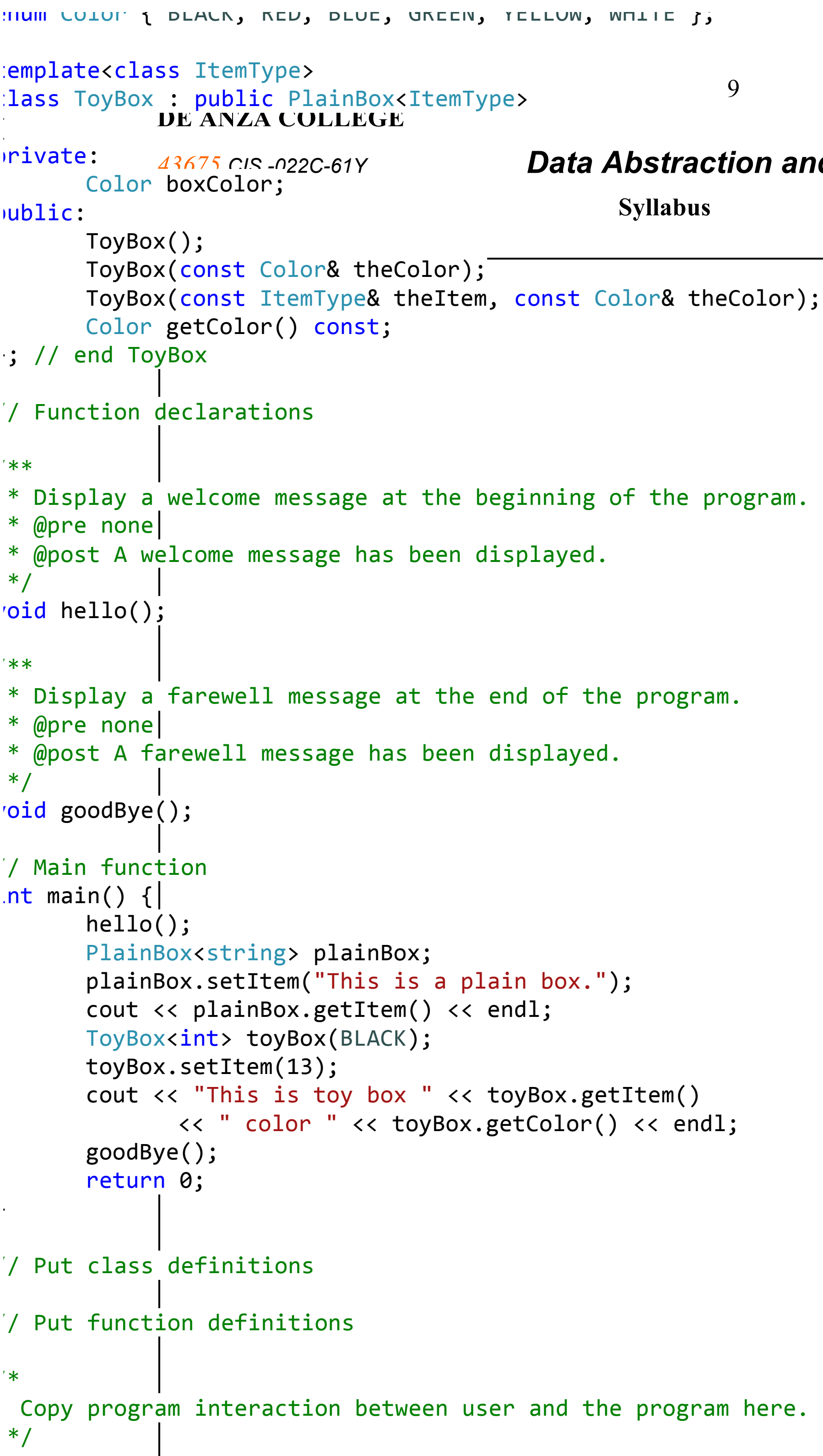
```
ription: Write a detailed description of WHAT this program does from a
ive.

FirstName LastName

ed: mm/dd/yyyy

ode / Visual Studio Express 2013 / Code::Blocks / ...

ements here
eam>
g>

 std;

ants

ations

 ItemType>
ace


void setItem(const ItemType& theItem) = 0;
ItemType getItem() const = 0;
terface

ItemType>
 public BoxInterface<ItemType>


field
 item;


lt constructor
();

eterized constructor
(const ItemType& theItem);

d to change the value of the data field
Item(const ItemType& theItem);

d to get the value of the data field
 getItem() const;
Box
```

DE ANZA COLLEGE **Spring, 2015**

43675 CIS -022C-61Y

Data Abstraction and Structures *(4 1/2 units)*

Syllabus



```
num Color { BLACK, RED, BLUE, GREEN, YELLOW, WHITE };

emplate<class ItemType>
lass ToyBox : public PlainBox<ItemType>

rivate:
      Color boxColor;
ublic:
      ToyBox();
      ToyBox(const Color& theColor);
      ToyBox(const ItemType& theItem, const Color& theColor);
      Color getColor() const;
; // end ToyBox

/ Function declarations

**
 * Display a welcome message at the beginning of the program.
 * @pre none
 * @post A welcome message has been displayed.
 */
oid hello();

**
 * Display a farewell message at the end of the program.
 * @pre none
 * @post A farewell message has been displayed.
 */
oid goodBye();

/ Main function
nt main() {
      hello();
      PlainBox<string> plainBox;
      plainBox.setItem("This is a plain box.");
      cout << plainBox.getItem() << endl;
      ToyBox<int> toyBox(BLACK);
      toyBox.setItem(13);
      cout << "This is toy box " << toyBox.getItem()
             << " color " << toyBox.getColor() << endl;
      goodBye();
      return 0;


/ Put class definitions

/ Put function definitions

*
 Copy program interaction between user and the program here.
 */
```

Tentative Lab Assignments (subject to adjustment during the course)

HW #1

a.) (Lists) Chapter 9. Programming Problem 8

Note: You must complete the implementation of LinkedList as specified in Chapter 9 and then use it for this exercise.

b.) (Stacks) Write a program that changes a decimal number to a hexadecimal number. The user of the program can enter a decimal number and get back its hexadecimal equivalent. If the user's input is not a decimal number, the program should give a message and let the user reenter the number.

HW #2

a.) (Queues) Chapter 13. Programing Problems 3. Implement the palindrome-recognition algorithm described in Section 13.2.2. Use ListQueue defined in Chapter 14 and LinkedStack defined in Chapter 7. You may need to complete the definition of those classes if you don't see the complete implementation. Test your function with these test strings: 1. an empty string; 2. a string that is a palindrome; 3. a string this is not a palindrome.

HW #3

a.) (BST) Chapter 16. Programing Problem 5. Create a BST and store the following data into it.

Lillian	Cox	3/5/1986
Kathy	Walker	10/10/1998
Gregory	Hughes	6/14/1989
James	James	4/26/1994
Ruby	Morris	4/30/1988
Jeremy	Russell	3/18/1980
Craig	Miller	10/29/1998
Teresa	Butler	12/29/1996
Jonathan	Washington	12/12/1996
Jesse	Gray	6/14/1991
Norma	Simmons	11/8/1990
Scott	King	5/6/1981
Andrea	Howard	3/30/1999
Andrew	Robinson	4/18/1993
Janet	Martin	5/26/1993
Carol	Baker	9/30/1988
Jessica	Watson	5/11/1982
Rose	Brown	4/8/1996
Stephanie	Taylor	9/19/1980
Lori	Murphy	5/18/1999
Betty	Gonzalez	7/1/1998
Barbara	Bell	12/19/1996
Ronald	Hernandez	3/27/1990
Kathryn	Evans	6/23/1990
Paul	Perez	6/30/1993

Debra	Morgan	5/19/1999
Angela	Griffin	6/10/1984
Albert	Harris	12/3/1996
Catherine	Phillips	4/3/1998
Heather	Foster	6/8/1995
Dennis	Smith	7/31/1995
Julia	Gonzales	3/18/1995
Stephen	Jackson	6/21/1987
Amy	Stewart	5/2/1992
Jimmy	Rivera	7/16/1999
Walter	Hall	10/25/1991
Larry	Hill	2/3/1984
Philip	Turner	3/3/1997
Daniel	Diaz	9/1/1997
Judith	Garcia	12/5/1991
Julie	Bailey	10/5/1981
Henry	Sanchez	3/25/1993
Gary	Henderson	2/3/1987

b.) Traverse the BST and display the names and birthdays in the alphabetic order of the names.
c.) Display the height of the BST.

HW #4

a.) (Priority Queues) Chapter 17. Programming Problem 6. Create a priority queue using the Heap_PriorityQueue and enter the to-do list as follows:

Send a birthday card to friend Julie. (High)
Start the research paper for world history. (Medium)
Finish reading Chapter 17 of Walls and Mirrors. (Medium)
Make plans for Saturday night. (Low)
Bring car to service. (High)
Make an appointment with my dentist. (High)
<Feel free to add more here>

b.) Display the contents of the to-do list in the order of priorities.

HW #5

a.) (Dictionaries) Create a dictionary using HashedDictionary and store the list of names and birthdays from HW#3 into the dictionary using the names as the keys.
b.) The program should provide a user interface for the user to ask for the information of a name. If the name exists in the dictionary, the information will be given; otherwise the user will see a proper message.

Team Projects (subject to adjustment during the course)

Each project will be done by one team consisting of 2 to 5 team members.

Objectives of Each Team

The team is to produce results that satisfy the requirements. The following are some of the outcomes to be produced by the team:

1. An **External Specification Document**. This document describes what the results will look like from the user's perspective in light of the requirements.
2. A **Design Document**. This document describes how to build the product. It answers questions such as: What are the components? What are the responsibilities of each component? How does the components interact with each other?
3. A **Test Plan**. This documents describes how to test the product and verify that the requirements are satisfied.
4. A **Presentation Document**. This is a PowerPoint document to be presented at the end of the project.
5. A **Demo**. This is a live demo to be shown during the presentation at the end of the project.

Team Members

Each team member's responsibilities are as follows:

1. Understand the requirements.
2. Cooperate and collaborate with other team members.
3. Contribute to external specification document if required.
4. Take initiatives to own and implement some of the components.
5. Ensure that the owned components do what are expected through unit tests.
6. Write test cases according to the test plan and produce test reports.

Team Leads

There will be one lead for each team. The lead's responsibilities are as follows:

1. Call a team meeting at least once a week for communications and collaboration.
2. Coordinate among the team members to ensure successful deliveries of the project.
3. Collect documents from the team members and provide weekly project report documents in **Word** to the instructor.
   First Report – External Specification Document
   Second Report – Design Document
   Third Report – Test Plan
   Fourth Report – Presentation Document
4. Create a PowerPoint for the demonstration of the project in Week #11.
5. Give the presentation and demo in Week #11.

Evaluation

A minus (-), a checkmark (√), or a plus (+) will be given to the team at each stage of the project. The assessments will be combined and converted to a score between 0 and 100 at the end of the project. The score will then be entered as each team member's team project score.

Potential Team Projects

43675 *CIS -022C-61Y* ***Data Abstraction and Structures*** *(4 1/2 units)*

Syllabus

1. (Quick Sort) Chapter 11. Programming Problems 1, 2, 3.
2. (Bank Simulation) Chapter 13. Programming Problems 6, 7
3. (Hospital Simulation) Chapter 17. Programming Problem 8.
4. (Dictionary) Chapter 19. Programming Problems 1, 3, 4.
5. (Dictionary) Chapter 21. Programming Problems 2, 3, 4.

-