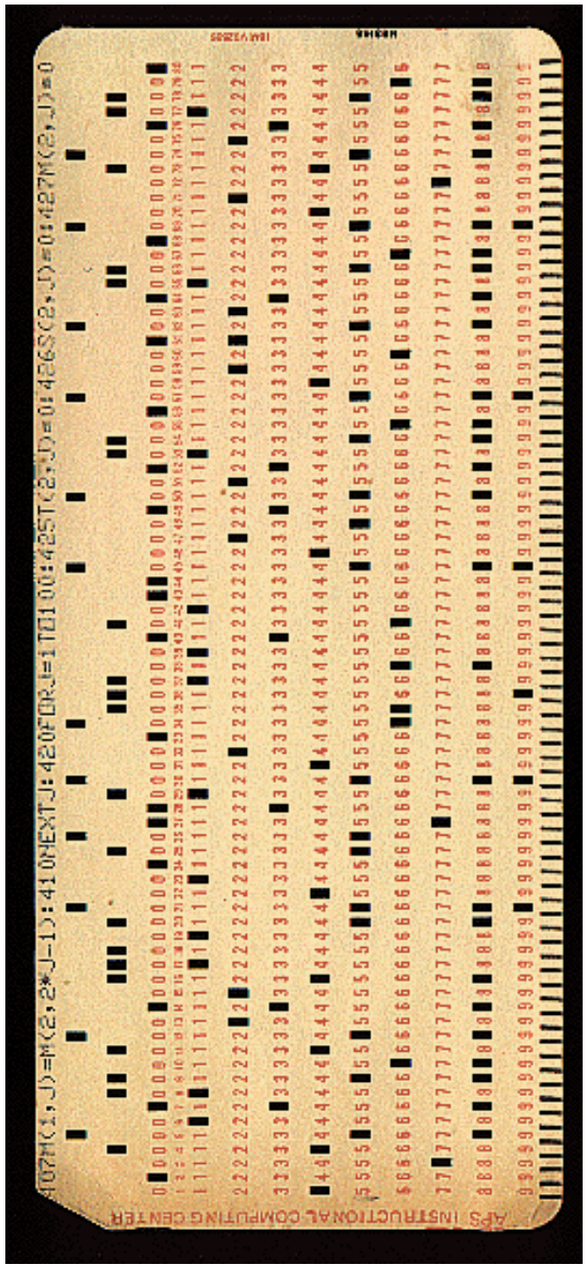
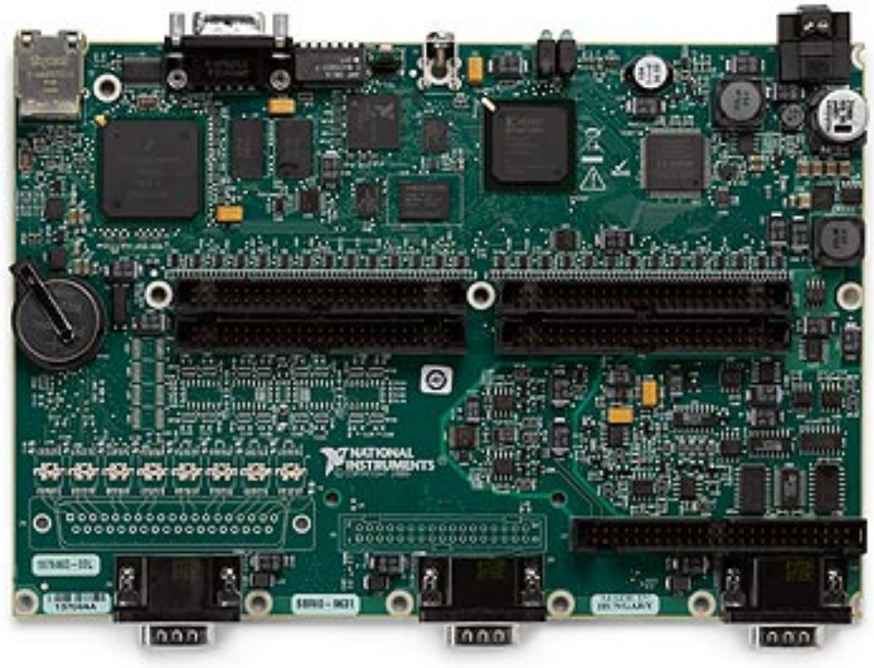
CSC 1052 – Algorithms & Data Structures II:

Introduction

Professor Henry Carter
Spring 2017

Programming





Address	Machine Language				Assembly Language
0000 0000	0000	0000	0000	0000	TOTAL .BLOCK 1
0000 0001	0000	0000	0000	0010	ABC .WORD 2
0000 0010	0000	0000	0000	0011	XYZ .WORD 3
0000 0011	0001	1101	0000	0001	LOAD REGD, ABC
0000 0100	0001	1110	0000	0010	LOAD REGE, XYZ
0000 0101	0101	1111	1101	1110	ADD REGF, REGD, REGE
0000 0110	0010	1111	0000	0000	STORE REGF, TOTAL
0000 0111	1111	0000	0000	0000	HALT

This course ...

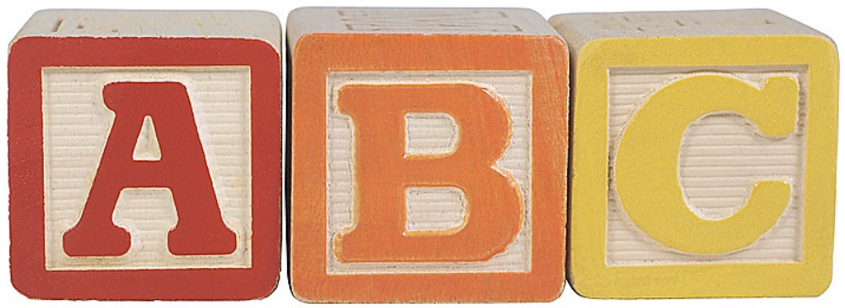
- We will investigate a series of data structures and algorithms designed to solve common problems and consider their broader applicability. Topics will include:
  - object-oriented design: inheritance, interfaces, polymorphism; problem analysis; recursion; abstract data types; dynamically linked structures; data structures: stacks, queues, collections, lists, trees.

Who cares?

- Computer Scientists/Engineers
  - Basic toolkit
- Scientific Computing
  - Simulation and data analysis
- Business management and economics
  - IT infrastructure and productivity tools
- Everyday people
  - Build problem-solving skills and automate common tasks

You need to understand ...

- Basic Java programming (CSC 1051 or equivalent)
- Be prepared to problem solve!



Goals

- My goal: to provide you with the tools to develop efficient code and solve common computing problems.
  - Data structures
  - Program design techniques
  - Basic algorithm design
- Pay-off: strong programming skills are critical for *any* computer science career (and beneficial for many other careers)
- Practice makes perfect! Use it or lose it!

Course Materials

- Website - I am maintaining the course website at
  - http://www.csc.villanova.edu/~carterh/Courses/csc1052/sp17/index.html
- Course assignments, slides, and other artifacts will be made available on the course website.
- Course textbook
  - *Object-Oriented Data Structures using Java*, Fourth Edition by Dale, Joyce, and Weems
  - Example code available on http://www.jblearning.com

Course Calendar

- The course calendar as all the relevant readings, assignments and test dates
- The calendar page contains electronic links to assignments and online papers assigned for course readings.
- Please check the website frequently for announcements and changes to the schedule. Students are responsible for any change on the schedule (I will try to make announcements in class).

Course Calendar

Below is the calendar for this semester course. This is the preliminary schedule, which may need to be altered as the semester progresses. It is the *responsibility of the students* to frequently check this web-page for schedule, readings, and assignment changes. As the professor, I will attempt to announce any change to the class, but this web-page should be viewed as authoritative. If you have any questions, please contact me (contact information is available at the course homepage).

Date	Topic	Assignments Due	Readings/Discussions (do readings before class)	Slides
08/25/16	Introduction		Course syllabus (link) Textbook, chapter 1 Assignment #0 - Introductions	
08/30/16	Algorithm Analysis	Assignment #0	Textbook, Chapter 2.1-2.2	
09/01/16	Nonrecursive Analysis		Textbook, Chapter 2.3	
09/06/16	Recursive Analysis		Textbook, Chapter 2.4-2.5	
09/08/16	Brute-force		Textbook, Chapter 3.1-3.2	
09/13/16	Exhaustive Search		Textbook, Chapter 3.4	
09/15/16	Tree Searching		Textbook, Chapter 3.5	
09/20/16	Insertion and Topological Sort		Textbook, Chapter 4.1-4.2	
09/22/16	Binary Search		Textbook, Chapter 4.4	
09/27/16	Variable Decrease Algorithms		Textbook, Chapter 4.5	
09/29/16	Merge and Quick Sort		Textbook, Chapter 5.1-5.2	
10/04/16	Further Divide-and-conquer		Textbook, Chapter 5.3-5.5	
10/06/16	**Mid-term Exam**			
10/11/16	**Fall Break -- No Class**			
10/13/16	**Fall Break -- No Class**			
10/18/16	Presorting and Gaussian Elimination		Textbook, Chapter 6.1-6.2	
10/20/16	Balanced Search Trees		Textbook, Chapter 6.3	
10/25/16	Representation Change		Textbook, Chapter 6.4-6.5	
10/27/16	Problem Reduction		Textbook, Chapter 6.6	
11/01/16	String Matching		Textbook, Chapter 7.1-7.2	
11/03/16	Dynamic Programming		Textbook, Chapter 8.1-8.2	
11/08/16	Dynamic Programming		Textbook, Chapter 8.3-8.4	
11/10/16	Prim's and Kruskal's Algorithms		Textbook, Chapter 9.1-9.2	
11/15/16	Dijkstra's and Huffman's Algorithm		Textbook, Chapter 9.3-9.4	
11/17/16	Iterative Improvement		Textbook, Chapter 10.1-10.2	
11/22/16	Iterative Improvement		Textbook, Chapter 10.3-10.4	
11/24/16	**Thanksgiving -- No Class**			
11/29/16	Lower Bound Arguments and Decision Trees		Textbook, Chapter 11.1-11.2	
12/01/16	P, NP, NP-Complete and Numerical Algorithms		Textbook, Chapter 11.3-11.4	
12/06/16	Backtracking and Branch and Bound		Textbook, Chapter 12.1-12.2	
12/08/16	Approximating NP-Hard Problems		Textbook, Chapter 12.3	
12/17/16	**Final Exam - 8:00am - 10:50am**			

Grading

- Grading in this class will be distributed as follows:
  40% Mid-term Exams
  20% Final Exam
  20% Quizzes Exercises Projects
  15% Labs
  5% Class Participation

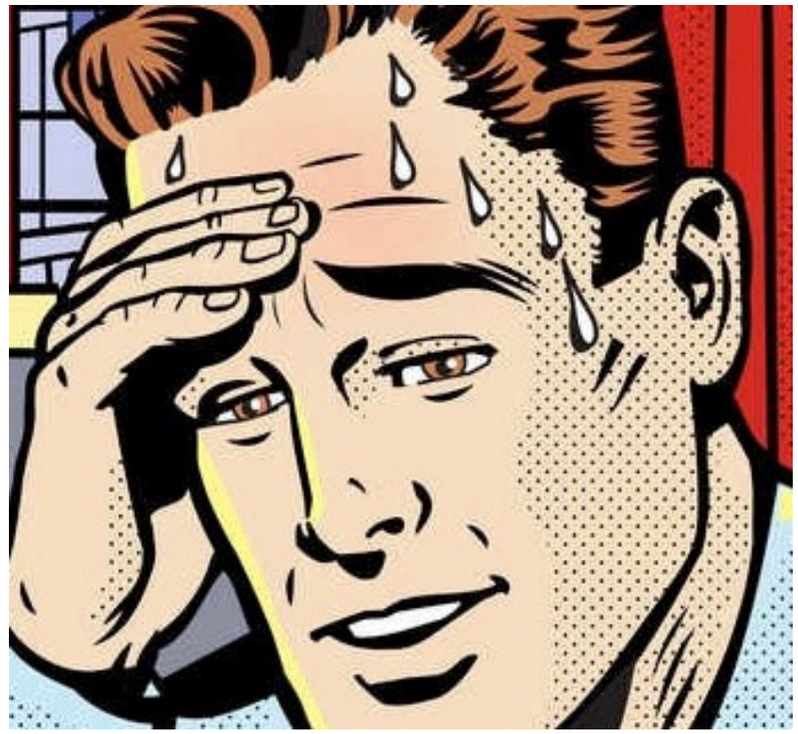
- *You get the grade that you earn, so be sure that you earn a grade you like.*

Readings

- There are readings from the textbook assigned for each lecture in this course. These assignments are intended to:
  - Support the lectures in the course (provide clarity)
  - Augment the lectures and provide a broader exposure to course topics.
- Students are required to do the reading!
- The material covered in class is *not enough* to fully understand the concepts of data structures and algorithm design. The reading is necessary to gain a complete understanding of the material and to do well on the homework and exams!

Homework

- Homeworks are to help solidify understanding
  - Ensure you understand the material
  - Help identify problem areas
- Bring your questions to me!
- Homework problems may be used on quizzes and exams…



Quizzes

- Pop quizzes may be given at the start of class to ensure that students are adequately preparing for each lecture
- These quizzes will be quick, so practice beforehand is critical to success

Course Projects

- Programming exercises requiring you to combine multiple course concepts into a single task
- Will require significant time and thought, so start early!
- NO collaboration allowed on these projects

Lateness

- All assignments is due at the beginning of class.
- Late assignments are assessed a 15% per-day late penalty, with a maximum of four days.
- Students with legitimate reasons should contact the professor before the deadline to apply for an extension.
  - Unless the problem is apocalyptic, don't give me excuses.

A Note on Academic Integrity

- As scientists and engineers, we must trust each other to make progress.
- Numerous examples exist to show the consequences of this breakdown.
  - Jan Hendrik Schon...
- Academic dishonesty, whether from cheating, copying, or through any other dishonest practice will not be tolerated.
  - I take this very personally - you should too.

Java: Review

- Built-in data types:
- Control flow statements:

Object-Oriented Programming

- Everything that isn't a basic data type is an object in Java
- Objects represent:
  - Information: objects have *attributes*
  - Behavior: objects have *responsibilities*
- Objects may map to real-world entities
- Objects are *modular*

Components of a Class

- An object is an instantiation of a class.
  - Alternately, a class defines the structure of its objects.
- A class definition includes:
  - Variables
  - Methods (constructors, observers, transformers)

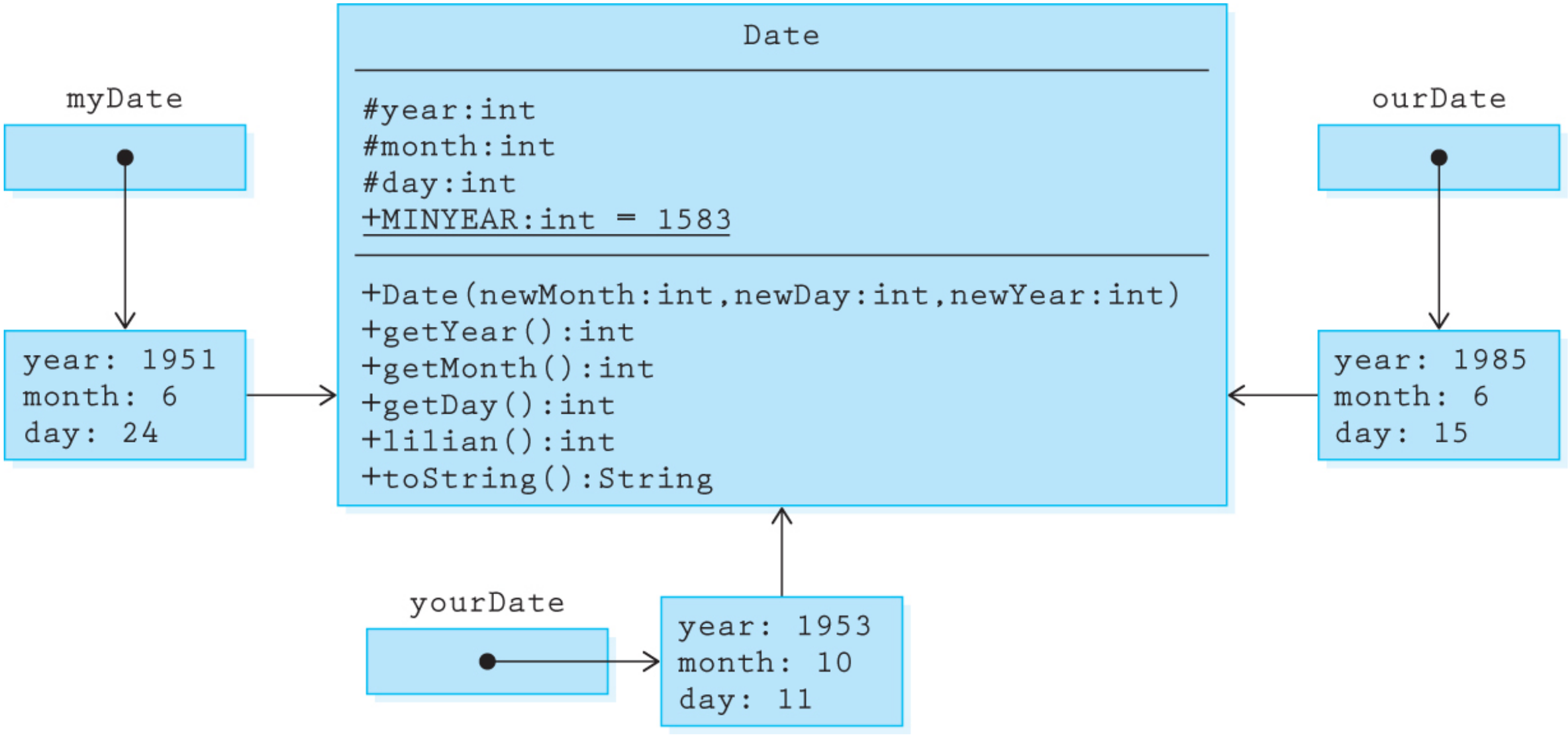
Example

```
package ch01.dates;
public class Date
{
  protected int year, month, day;
  public static final
      int MINYEAR = 1583;

  // Constructor
  public Date(int newMonth,
              int newDay,
              int newYear)
  {
    month = newMonth;
    day = newDay;
    year = newYear;
   }

  // Observers
  public int getYear()
  {
    return year;
  }

  public int getMonth()
  {
    return month;
  }

  public int getDay()
  {
    return day;
  }

  public int lilian()
  {
    // Returns the Lilian Day Number
    // of this date.
    // Algorithm goes here.
  }

  @Override
  public String toString()
  // Returns this date as a String.
  {
    return(month + "/" + day
                 + "/" + year);
  }
}
```

The Unified Method

- Use-case driven
- Iterative and incremental
- Architecture-centric

Class Diagrams



Putting it all together

- Designate one class as the *main* class
  - The class containing the *main* method is a Java "application"
- Combine Java source code into a compiled program
  - Java bytecode
- The Java Virtual Machine (JVM) executes the program

Save the (Integrated Development) Environment!

- Development environments are numerous and varying
- Using a variety of IDEs will help you identify features you like or dislike
- We will be using the Eclipse IDE in this course
- Start thinking about optimizing your IDE now!

Next Time...

- Dale, Joyce, Weems Chapter 1.1-1.5
  - Remember, you need to read it BEFORE you come to class!
- Download and install Eclipse IDE
  - http://www.eclipse.org/
- Check out Hacker Rank
  - https://www.hackerrank.com/