CS 240

Fall 2015

Prof. John Bowers

https://w3.cs.jmu.edu/bowersjc/classes/fall15/cs240/

A Word Game

Write the following words:

SPIT
NOT
SO
FAT
FOP
AS
IF
IN
PAN

2 Players

Rules:

1. Play alternates between players.

2. Each turn the current player selects one previously unselected word.

3. The first player to select three words all sharing a common letter wins.

A Word Game

Write the following words:

~~SPIT~~
~~NOT~~
SO
~~EAT~~
~~FOR~~
~~AS~~
~~IF~~
IN
~~PAN~~

2 Players

Rules:

1. Play alternates between players.

2. Each turn the current player selects one previously unselected word.

3. The first player to select three words all sharing a common letter wins.

Any Winning Strategy or **Algorithm**?

The organization of information makes a huge difference in how easy it is to perform tasks with that information.

Data Structures: How you organize your data

abstraction
abstraction
abstraction

Ex: Arrays, Linked Lists, Lists

- Key Concepts
  - **Abstract Data Type (ADT)** vs. **Implementation**
  - **Information Hiding**
  - **Levels of Abstraction**

Main CS 240 Data Structure Topics

- Stacks and Queues
- Lists
- Hashtables
- Maps
- Trees
- and more…

Algorithms: How you solve problems (using data structures)

Hint: You've already been using algorithms.

Big Goal: Efficiency

But what is efficiency?

Main CS240 Algorithm Topics

- Recursion
- Asymptotic analysis
- Big-O Notation
- searching
- sorting

Strong ties between Data Structures and Algorithms

- Data structure operations are themselves algorithms.
- Algorithms use data structures.
- The efficiency of an algorithm is effected by what data structures are used.

Key Concepts

- **Abstract Data Types (ADT)**
  - Generic classes of data structures
  - Often provided with a languages standard library
  - Two goals: efficiency and reusability
- **Algorithm analysis**
  - How efficient? (Again, what is efficiency?)
  - Comparing two algorithms: which is faster?
  - Empirical (test it) and analytical (math)

Goals

- Describe, implement, and analyze common data structures and algorithms.
- Choose appropriate structures and algorithms to solve real-world problems.
- Be able to *communicate* about data structures and algorithms at the appropriate level of abstraction.
- Secondary goal: become familiar with a language that is not Java.

Meta-Goals

- Learn *how to learn* programming languages.

  WARNING: We will not give you everything you need to know about C.

- Get you into the mindset of try-it-and-see.

The C Language

- Invented by Dennis Ritchie, early 1970s at Bell Labs for development of Unix OS.

- Like Java, it is a general purpose, imperative, structured, programming language with static scoping and typing (CS 440)

    - Unlike Java, NOT Object Oriented, NOT automatically garbage collected

- Usually compiled to machine code.

- Used for nearly 50 years.

C Example

```c
#include <stdio.h>

int main()
{
    int value = 1;
    for (int x = 0; x <= 10; x++) {
        printf("2 ^ %d = %d\n", x, value);
        value *= 2;    // calculate next power of 2
    }
    return 0;
}
```

Colors:

keywords
local variables
string literal
comment

C Programming

- **Y'all are our Guinea Pigs.**
- But Why C?
  - ***Familiarity***. You know Java. The non-OO parts of Java are based on C.
  - C is NOT object oriented, you are going to have to really get nitty-gritty with it.
  - Simple, mature (faster code), ubiquitous (high likelihood you will use it in your career).
- **Aligns well with the purposes of this class.**
  - You have to manually allocate memory. (What does **new** do in Java, exactly?)
  - Static Typing (compiler helps you with debugging)
  - **Clear distinction** between data and pointers.

Code Style

- There are many different style guides for C: https://en.wikipedia.org/wiki/Indent_style
  - My preferred style is 1TBS ("the one true brace style," based on K&R)
  - Other instructors at JMU use the GNU style
- **The Key is Consistency and Readability: Who is your audience?**
- Keep Code Clean
  - **If I have trouble reading it, I will deduct points**
  - If your code style and/or whitespace is inconsistent, I will deduct points
- Include documentation
  - If I have trouble understanding it, I will deduct points

Bad Style

```
                    #include <stdio.h>
                int
main()          {       int v = 1;
                        for (int x = 0;
x <= 10; x++) {
                                printf(
"2 ^ %d = %d\n”

,
x, v);v *= 2;    }                      return 0;}
```

Colors:

keywords

local variables

string literal

comment

Success in Learning C

You will need to work on C outside of just the assignments and labs.

- We do not have enough time to cover everything you need to know in the labs.
- Quickly learning new languages is an important meta skill.
- I recommend spending 30 min to an hour a day working at C.
- Learn how to ask for help. (Office hours, Piazza, Each other)
- Hybrid Lecture / Labs
- “Sanity Check” Programming project (PA0)

Syllabus Highlights

Course website: https://w3.cs.jmu.edu/bowersjc/classes/fall15/cs240/

- Syllabus
- Calendar
- Assignments
- Resources

It is your responsibility to know and read the syllabus.

Canvas: - Grades and online quizzes. - Project Submission - Private files
Piazza: http://canvas.jmu.edu

Textbook

- Required textbook: “Open Data Structures”
  - Open-source textbook by Pat Morin
  - Available as a PDF from http://opendatastructures.org/
  - Available as a coursepack from JMU bookstore
  - Available as a paperback online (see link in syllabus: order **now**!)
- Recommended book: "The C Programming Language"
  - Brian Kernighan and Dennis Ritchie (creator of C)
  - Available on Safari Books through the library

Policies

- Reading quizzes are due by 9am on the day they are to be covered.
    - One-time exception for today's quiz (due tomorrow)
- Class attendance is **strongly** recommended
    - If the class periods are not worth attending, tell me so I can make them better!
- Slides posted on website.
- Please silence cell-phones during class.
    - Be respectful with laptop and tablet usage.

Programming Projects

- Submit programming projects as specified in project description.
  - No thumb drives, CDs, or emails (unless requested)
- Project grading will be based on automated test results.
  - Public vs. **Private** tests.
  - Some portion of the grade (usually 10-20%) for code **style and documentation.**
  - Late submissions up to 72 hours late, with a 10% penalty each 24 hour period.

Group Work

- The JMU Honor Code applies on ALL assignments
  - We **will** use software to detect plagiarism
  - Violations may be sent to the honor council
  - See relevant section in the syllabus
- All submitted code must be YOUR work entirely
  - *But*, you may work in groups to discuss assignments (in fact, **I encourage this**), but do NOT share code!

Grades

- Homework Assignments: 30%
- Programming Projects: 25%
- Midterm Exams: 30%
- Final Exam: 15%

Exams

- Exams will be held in ISAT 243.
- Final exam times are on the website.
- If you ask for a re-grade, I may re-grade the entire assignment This applies to homework and projects, too.
- If you have to miss a due date or exam because of an excused absence, let me know ASAP.

Involvement Ops

- JMU Unix Users' Group
  - Club of Unix/GNU/Linux fans who gather for fun and tech talk
  - Tutorial series throughout semester
    - Linux InstallFest on Wed, Sept. 2 at 7pm in ISAT/CS 259 (nTelos Room) – repeated on Tue, Sept. 8 (same time and place)
    - HIGHLY RECOMMENDED for CS 240 students

- ACM Competitive Programming Club
  - Club of students who enjoy programming and problem solving
  - Meetings and contests throughout semester
    - First meeting on Fri, Sept. 4 at 2:30pm in ISAT/CS 143
    - Repeats every Friday (same time and place)
    - Can be taken as a course for one credit if you wish (register for CS 280)
    - HIGHLY RECOMMENDED for CS 240 students

Homework

- Complete course survey by tomorrow (on Canvas)
- Complete first reading quiz by tomorrow (on Canvas)
- If you have a personal computer, install a C compiler and a text editor (see "Resources" page on website)
    - Alternatively, go into ISAT 248 and familiarize yourself with the software on the lab computers
    - Change the code snippet from the earlier slide to calculate Fibonacci numbers rather than powers of two
- Consider attending Linux InstallFest on Wednesday
    - They will help you set up a development environment for this class

Questions

- Piazza! Piazza! Piazza!
- Email: bowersjc
    - Do not expect response in under 24 hours
- Office
    - Office hours MW 1:30-4p

Have a great semester