CS1010 Programming Methodology
A beginning in problem solving in Computer Science

Aaron Tan
http://www.comp.nus.edu.sg/~cs1010/
20 July 2015
Announcements

This document is available on the CS1010 website

Announcements

Choosing CS1101S over CS1010

- URL: https://register.comp.nus.edu.sg/UGOffice4
- Deadline: 23 July 2015, Thursday, 6pm
- Default: CS1010 (No action needed if you decide to take CS1010)
- No bidding queue in CORS for both CS1010 and CS1101S
- Students from IS Dept are precluded from taking CS1101S

DDP students in CS and Maths/Applied Maths will be pre-allocated CS1101S

Diploma holders who are exempted CS1010 will be pre-allocated CS1020

For more details, please contact SoC Undergraduate Office @ COM1, Level 2, Room 19
Computer Science Curricula 2013

- Ironman draft identifies 18 knowledge areas

- AL - Algorithms and Complexity
- AR - Architecture and Organization
- CN - Computational Science
- DS - Discrete Structures
- GV - Graphics and Visual Computing
- HCI - Human-Computer Interaction
- IAS – Security and Info Assurance
- IM - Information Management
- IS - Intelligent Systems
- NC – Networking and Communication
- OS - Operating Systems
- PBD – Platform-based Development
- PD – Parallel and Distributed Computing
- PL - Programming Languages
- SDF – S/W Dev. Fundamentals
- SE - Software Engineering
- SF – Systems Fundamentals
- SP - Social and Professional Issues

P = NP ?

O(n^2)

[CS1010: A beginning in problem solving in CS]
Introduces the fundamental concepts of problem solving by computing and programming using an imperative programming language.

Outcomes

- Solve simple algorithmic problems
- Write good small programs

C as a tool

Not just about C
Programming

Language constructs

Problem solving

Coding
A C Program (welcome.c)

// Author: Aaron Tan
// Purpose: Ask for user's name and display a welcome message.

#include <stdio.h>

int main(void) {
    char name[20];
    printf("What is your name? ");
    scanf("%s", name);
    printf("Hi %s.\n", name);
    printf("Welcome to CS1010!\n");
    return 0;
}
Problem Solving Process

**Determine problem features**

**Write algorithm**

**Produce code**

**Check for correctness and efficiency**

---

**Analysis**

**Design**

**Implementation**

**Testing**

Rethink as appropriate
Your Friendly CS1010 Lecturers

<table>
<thead>
<tr>
<th>Prof. Wynne Hsu</th>
<th>Dr Zhao Jin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office: COM2 #03-05</td>
<td>Office: COM2 #02-50</td>
</tr>
<tr>
<td><a href="mailto:whsu@comp.nus.edu.sg">whsu@comp.nus.edu.sg</a></td>
<td><a href="mailto:zhaojin@comp.nus.edu.sg">zhaojin@comp.nus.edu.sg</a></td>
</tr>
</tbody>
</table>
Your SG and DG will be pre-allocated to you.

All groups cover the SAME syllabus, and have common tests, and all students are graded together as a whole. Grading NOT by bell curve.
Workload (4 MCs)

- **Lectures:**
  - 3 hours/week in a lab setting.

- **Discussion sessions:**
  - 2 hours/week in a lab setting.

- **Continual assessment:**
  - Take-home lab assignments
  - 2 Practical Exams (Saturdays)
  - Term Test (Saturday)
  - Final Exam
A Peek at a Lecture Session (1/2)

Lecturer’s screen is broadcast to every student’s monitor.

Interacting with students always makes me happy.
A Peek at a Lecture Session (2/2)

Explaining how to edit and compile a program.

Discussing MasterMind.
Welcome to CS1010!

Quick access to useful links:
- CodeCrunch website
- IVLE

Please read the IVLE discussion forum when the semester commences.

Things to prepare before your first lecture

Dear CS1010 students,

To have a pleasant experience in your first lecture, please note the following:
- Bring along/remember your NUSNET account-id and password, so that you can log into the computer.
- Create your UNIX account before you come for class. The website is [https://mysoc.comp.nus.edu.sg/~newacct/](https://mysoc.comp.nus.edu.sg/~newacct/)
- You may also refer to [https://docs.comp.nus.edu.sg/node/1517](https://docs.comp.nus.edu.sg/node/1517) for other related information, such as what you could do if you forget your UNIX account password.
- Bring your matriculation card, as you may need it to gain access into the programming labs. (Note that the card readers may be deactivated in the first week of class for the convenience of our freshmen. In this case, you can get into the lab without the need of your card. But still, it is better to bring your card along just in case.)
- If your card does not work, please send an email indicating your matriculation number to smartcardop@comp.nus.edu.sg to inform them of the problem.
- You may refer to the “Venues” section in the webpage [http://www.comp.nus.edu.sg/~cs1010/1_module_info/sched.html](http://www.comp.nus.edu.sg/~cs1010/1_module_info/sched.html) to find out where your lab is. Please check the venue of your lecture and arrive there on time.

Thanks and the CS1010 team look forwards to meeting you!

Aaron Tan
CS1010 coordinator
IVLE

https://ivle.nus.edu.sg

Watch out for announcements

Participate in the forums

Multimedia videos
Mid-Semester Review
Topics in C covered so far

- **Basic C program structure**
  - main() function
  - Variable declarations
  - Data types (int, float, double, char)
  - Arithmetic operations (+, -, *, /, %)
  - Input/output functions (scanf(), printf())

- **Preprocessor directives**
  - #include
  - #define

- **Control structures**
  - Sequential statements
  - Selection statements
    - Relational operators (<, <=, >, >=, ==, !=)
    - Logical operators (&&, ||, !)
    - Conditional operator (? :)
    - Integer as boolean
    - if, if-else, switch
  - Repetition statements
    - while, do-while, for

- **Functions**
  - Return type
  - Parameters
  - Function prototypes
  - Scope of variables/parameters

- **Pointers**

- **Arrays**
Mid-Semester Review

Topics in C

Programming environment/tools
- Operating system: UNIX
- Editor: vim
- Debugger: gdb

Program development
- Writing pseudocodes
- Edit – compile – execute” cycle
- Step-wise refinement
- Hand-tracing codes
- Incremental coding
- Testing
- Debugging

Problem solving
- Class exercises
- Practice exercises
- Lab assignments
Algorithmic Problem Solving #1: Coin Change

- Given these coin denominations: 1¢, 5¢, 10¢, 20¢, 50¢, and $1, find the smallest number of coins needed for a given amount. You do not need to list out what coins are used.

- Example 1: For 375 cents, 6 coins are needed.
- Example 2: For 543 cents, 10 coins are needed.
Algorithmic Problem Solving #1: Coin Change

Algorithm:
in:amt (in cents)
out:coins
coins \leftarrow 0
coins += \text{amt}/100; \quad \text{amt} = \text{remainder} \text{ of} \text{\text{amt}}/100;
coins += \text{amt}/50; \quad \text{amt} = \text{remainder} \text{ of} \text{\text{amt}}/50;
coins += \text{amt}/20; \quad \text{amt} = \text{remainder} \text{ of} \text{\text{amt}}/20;
coins += \text{amt}/10; \quad \text{amt} = \text{remainder} \text{ of} \text{\text{amt}}/10;
coins += \text{amt}/5; \quad \text{amt} = \text{remainder} \text{ of} \text{\text{amt}}/5;
coins += \text{amt}/1; \quad \text{amt} = \text{remainder} \text{ of} \text{\text{amt}}/1;
print coins
Algorithmic Problem Solving #2: Maximum Sum of Path in a Pyramid

Figure 1. (a) A pyramid of integers. (b) A path with sum of 13. (c) A path with sum of 18.
Maximum Sum of Path in a Pyramid

5
8  6
10 11 14
15 13 18 17
### Maximum Sum of Path in a Pyramid

```c
int maxPathValue(int arr[][MAX_ROWS], int size) {
    int r, c, max;

    for (r = 1; r < size; r++) {
        arr[r][0] += arr[r-1][0]; // left-most item
        for (c = 1; c < r; c++) {
            if (arr[r-1][c-1] > arr[r-1][c])
                arr[r][c] += arr[r-1][c-1];
            else
                arr[r][c] += arr[r-1][c];
        }
        arr[r][r] += arr[r-1][r-1]; // right-most item
    }

    // find maximum in last row
    max = arr[size-1][0];
    for (c = 1; c < size; c++)
        if (arr[size-1][c] > max)
            max = arr[size-1][c];

    return max;
}
```

---

Search for largest value in last row.
Maximum Sum of Path in a Pyramid

Why not from bottom to top?

```c
int maxPathValue(int arr[][MAX_ROWS], int size) {
    int r, c;
    for (r = size-2; r >= 0; r--) {
        for (c = 0; c <= r; c++) {
            if (arr[r+1][c] > arr[r+1][c+1])
                arr[r][c] += arr[r+1][c];
            else
                arr[r][c] += arr[r+1][c+1];
        }
    }
    return arr[0][0];
}
```
Algorithmic Problem Solving #3: Mad Scientist

- A mad scientist wishes to make a chain out of plutonium and lead pieces. There is a problem, however. If he places two pieces of plutonium next to each other…

  In how many ways can he safely construct a chain of length 6?

- General case: What about length \( n \)?
## Algorithmic Problem Solving #3: Mad Scientist

<table>
<thead>
<tr>
<th>Length</th>
<th>#ways</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>?</td>
</tr>
<tr>
<td>$n$</td>
<td></td>
</tr>
</tbody>
</table>
Algorithmic Problem Solving #4: Sudoku

<table>
<thead>
<tr>
<th>5</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>2</td>
<td>1</td>
<td>9</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>1</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>9</td>
</tr>
</tbody>
</table>
Algorithmic Problem Solving #5: MasterMind (1/2)

- **Sink**: Correct colour, correct position
- **Hit**: Correct colour, wrong position

<table>
<thead>
<tr>
<th>Secret code</th>
<th>Sinks</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess #1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Guess #2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Guess #3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Guess #4</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Secret code</th>
<th>Sinks</th>
<th>Hits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess #1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Guess #2</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Guess #3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Guess #4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Algorithmic Problem Solving #5: MasterMind (2/2)

- 6 colours:
  - R: Red
  - B: Blue
  - G: Green
  - Y: Yellow
  - C: Cyan
  - M: Magenta

- Given a secret code (secret) and a player’s guess (guess), how do we compute the number of sinks and hits?
CS1010 versus CS1101S

The differences

<table>
<thead>
<tr>
<th></th>
<th>CS1010</th>
<th>CS1101S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MC</strong></td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>Imperative PL (C)</td>
<td>Functional PL (Javascript)</td>
</tr>
<tr>
<td><strong>Enrolment</strong></td>
<td>≈ 160</td>
<td>≈ 110</td>
</tr>
<tr>
<td><strong>Lecture mode</strong></td>
<td>In lab</td>
<td>In LT/seminar room</td>
</tr>
</tbody>
</table>

The similarities

- Small-group teaching
- Assume no programming background
- Expect students to put in much effort; independent learning
What to Prepare Before Class Starts?

- Check out CS1010 website http://www.comp.nus.edu.sg/~cs1010
  - Learn UNIX
  - Learn vim
Attitude is Everything

- Your attitude, not your aptitude, will determine your altitude.
- If you think you can, you can. If you think you cannot, you are right.
- Don’t complain about heavy workload.
- Work hard, REALLY hard!
We are doing everything we can to help you

- Exercises during sectional teaching
- Exercises during discussion sessions
- Practice exercises on CodeCrunch
- On-line quizzes
- IVLE forums
- Help sessions
Announcements

Choosing CS1101S over CS1010

- URL: https://register.comp.nus.edu.sg/UGOffice4
- Deadline: 23 July 2015, Thursday, 6pm
- Default: CS1010 (No action needed if you decide to take CS1010)
- No bidding queue in CORS for both CS1010 and CS1101S
- Students from IS Dept are precluded from taking CS1101S

DDP students in CS and Maths/Applied Maths will be pre-allocated CS1101S

Diploma holders who are exempted CS1010 will be pre-allocated CS1020

For more details, please contact SoC Undergraduate Office @ COM1, Level 2, Room 19
Announcements

This document is available on the CS1010 website

Have a GREAT TIME in School of Computing!

{ Leading the world with Asia's best }