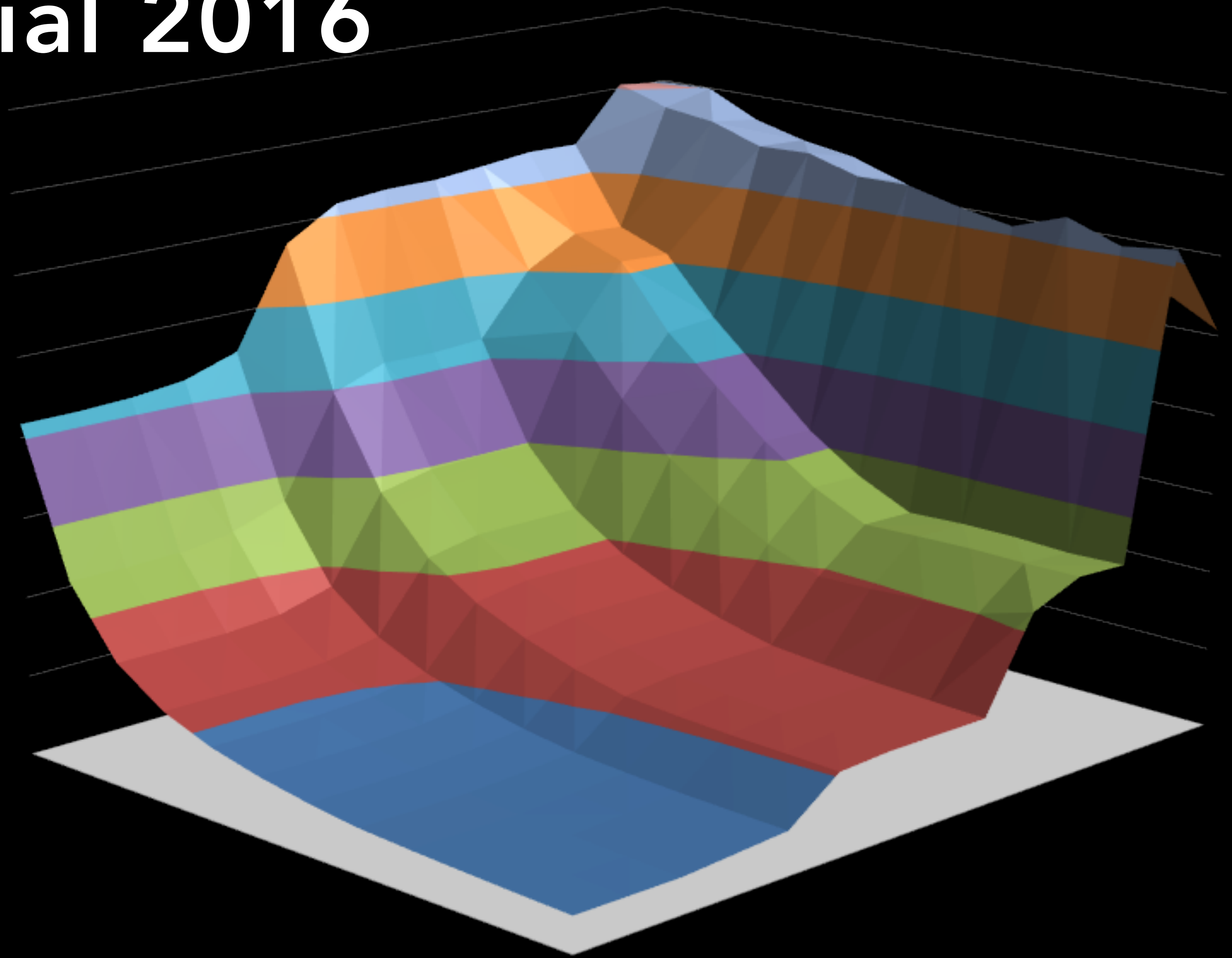
INSTRUCTOR: *JIANGFANG YI* TUTOR: *YILONG LI*

Introduction to Computer System

Tutorial 2016



WELCOME TO ICS

- **Instructor**
  - Prof. Jiangfang Yi
  - Mail: yijiangfang@mprc.pku.edu.cn

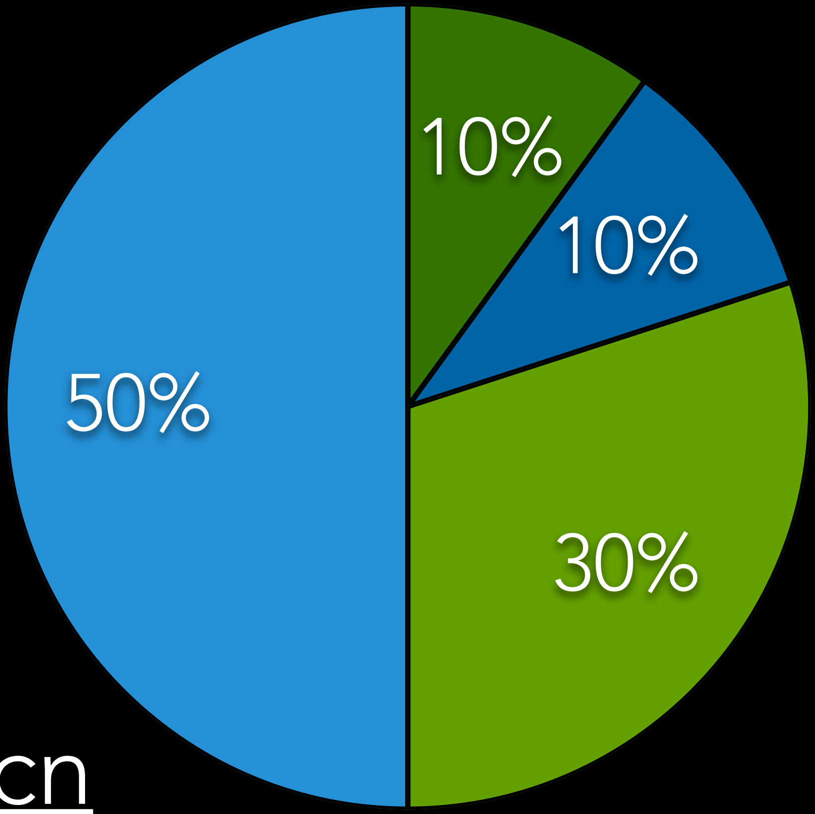
Illustrated by Shuai YANG

WELCOME TO ICS

- **Tutor: Yilong Li**
  - 4th year undergraduate
  - Mail: yilong@pku.edu.cn
  - WeChat: 13718037213

SCORE

- Discussion and review: 10%
- Homework: 10%
    - http://115.28.88.148/ics
- Lab: 30%, https://ics16.pku.edu.cn
- Mid-term & Final exam: 20%+30% (?)



REQUIREMENT

- **Review**
  - Time: 25 minutes / lecture
    - 20 min for PPT review
    - 5 min for problems, …
  - Send the presentation (PPT/PDF) before Thursday
  - (tentative) Other students' remark may affect your score

REQUIREMENT

- **Homework**
  - Submission: http://115.28.88.148
  - Only PDF/C/CPP/tar.gz are accepted
  - Using your student-id and PIN code
  - Submit before deadline
    - Deadlines will be emailed and be announced in the WeChat group
    - Late submission will be penalized

REQUIREMENT

- **Lab**
  - Do it yourself
  - Submit before deadline

Csdn.net
GitHub

Academic Integrity Of Carnegie Mellon Univ.

Cheating/Plagiarism: Description

- **Unauthorized use of information**
  - Borrowing code: by copying, retyping, **looking at** a file
  - Describing: verbal description of code from one person to another.
  - Searching the Web for solutions
  - Copying code from a previous course or online solution
  - Reusing your code from a previous semester (here or elsewhere)
    - If specific to 213/513, and you received credit

Cheating/Plagiarism: Description (cont.)

- **Unauthorized supplying of information**
  - Providing copy: Giving a copy of a file to someone
  - Providing access:
    - Putting material in unprotected directory
    - Putting material in unprotected code repository (e.g., Github)
  - Applies to this term and the future
    - There is no statute of limitations for academic integrity violations

Cheating/Plagiarism: Description

- **What is NOT cheating?**
  - Explaining how to use systems or tools
  - Helping others with *high-level* design issues
  - Using code supplied by us
  - Using code from the CS:APP web site

- **See the course syllabus for details.**
  - Ignorance is not an excuse

Cheating: Consequences

- **Penalty for cheating:**
  - Best case: -100% for assignment
    - You would be better off to turn in nothing
  - Worst case: Removal from course with failing grade
    - This is the default
  - Permanent mark on your record
  - Loss of respect by you, the instructors and your colleagues
  - If you do cheat – come clean asap!

- **Detection of cheating:**
  - We have sophisticated tools for detecting code plagiarism
  - In Fall 2015, 20 students were caught cheating and failed the course.
    - Some were **expelled** from the University
  - In January 2016, 11 students were penalized for cheating violations that occurred as far back as Spring 2014.

- **Don't do it!**
  - Manage your time carefully
  - Ask the staff for help when you get stuck

Some Concrete Examples:

This is Cheating:

- Searching the internet with the phrase 15-213, 15213, 213, 18213, malloclab, etc.
  - That’s right, just entering it in a search engine
- Looking at someone’s code on the computer next to yours
- Giving your code to someone else, now or in the future
- Posting your code in a publicly accessible place on the Internet, now or in the future
- Hacking the course infrastructure

This is OK (and encouraged):

- Googling a man page for fputs
- Asking a friend for help with gdb
- Asking a TA or course instructor for help, showing them your code, …
- Looking in the textbook for a code example
- Talking about a (high-level) approach to the lab with a classmate

How it Feels: Student and Instructor

- Fred is desperate. He can't get his code to work and the deadline is drawing near. In panic and frustration, he searches the web and finds a solution posted by a student at U. Oklahoma on Github. He carefully strips out the comments and inserts his own. He changes the names of the variables and functions. Phew! Got it done!
- The course staff run checking tools that compare all submitted solutions to the solutions from this and other semesters, along with ones that are on the Web.
  - Remember: We are as good at web searching as you are
- Meanwhile, Fred has had an uneasy feeling: Will I get away with it? Why does my conscience bother me?
- Fred gets email from an instructor: "Please see me tomorrow at 9:30 am."
  - Fred does not sleep well that night

How it Feels: Student and Instructor

- **The instructor feels frustrated. His job is to help students learn, not to be police. Every hour he spends looking at code for cheating is time that he cannot spend providing help to students. But, these cases can't be overlooked**
- **At the meeting:**
  - Instructor: "Explain why your code looks so much like the code on Github."
  - Fred: "Gee, I don't know. I guess all solutions look pretty much alike."
  - Instructor: "I don't believe you. I am going to file an academic integrity violation."
    - Fred will have the right to appeal, but the instructor does not need him to admit his guilt in order to penalize him.
- **Consequences**
  - Fred may (most likely) will be given a failing grade for the course
  - Fred will be reported to the university
  - A second AIV will lead to a disciplinary hearing
  - Fred will go through the rest of his life carrying a burden of shame
  - The instructor will experience a combination of betrayal and distress

A Scenario: Cheating or Not?

Alice is working on malloc lab and is just plain stuck. Her code is seg faulting and she doesn't know why. It is only 2 days until malloc lab is due and she has 3 other assignments due this same week. She is in the cluster.

Bob is sitting next to her. He is pretty much done.

Sitting next to Bob is Charlie. He is also stuck.

- 1. Charlie gets up for a break and Bob makes a printout of his own code and leaves it on Charlie's chair.
  - Who cheated: Charlie? Bob?
- 2. Charlie finds the copy of Bob's malloc code, looks it over, and then copies one function, but changes the names of all the variables.
  - Who cheated: Charlie? Bob?

Another Scenario

Alice is working on malloc lab and is just plain stuck. Her code is seg faulting and she doesn't know why. It is only 2 days until malloc lab is due and she has 3 other assignments due this same week. She is in the cluster.

Bob is sitting next to her. He is pretty much done.

Sitting next to Bob is Charlie. He is also stuck.

- 1. Bob offers to help Alice and they go over her code together.
  - Who cheated: Bob? Alice?
- 2. Bob gets up to go to the bathroom and Charlie looks over at his screen to see how Bob implemented his free list.
  - Who cheated: Charlie? Bob?

Another Scenario (cont.)

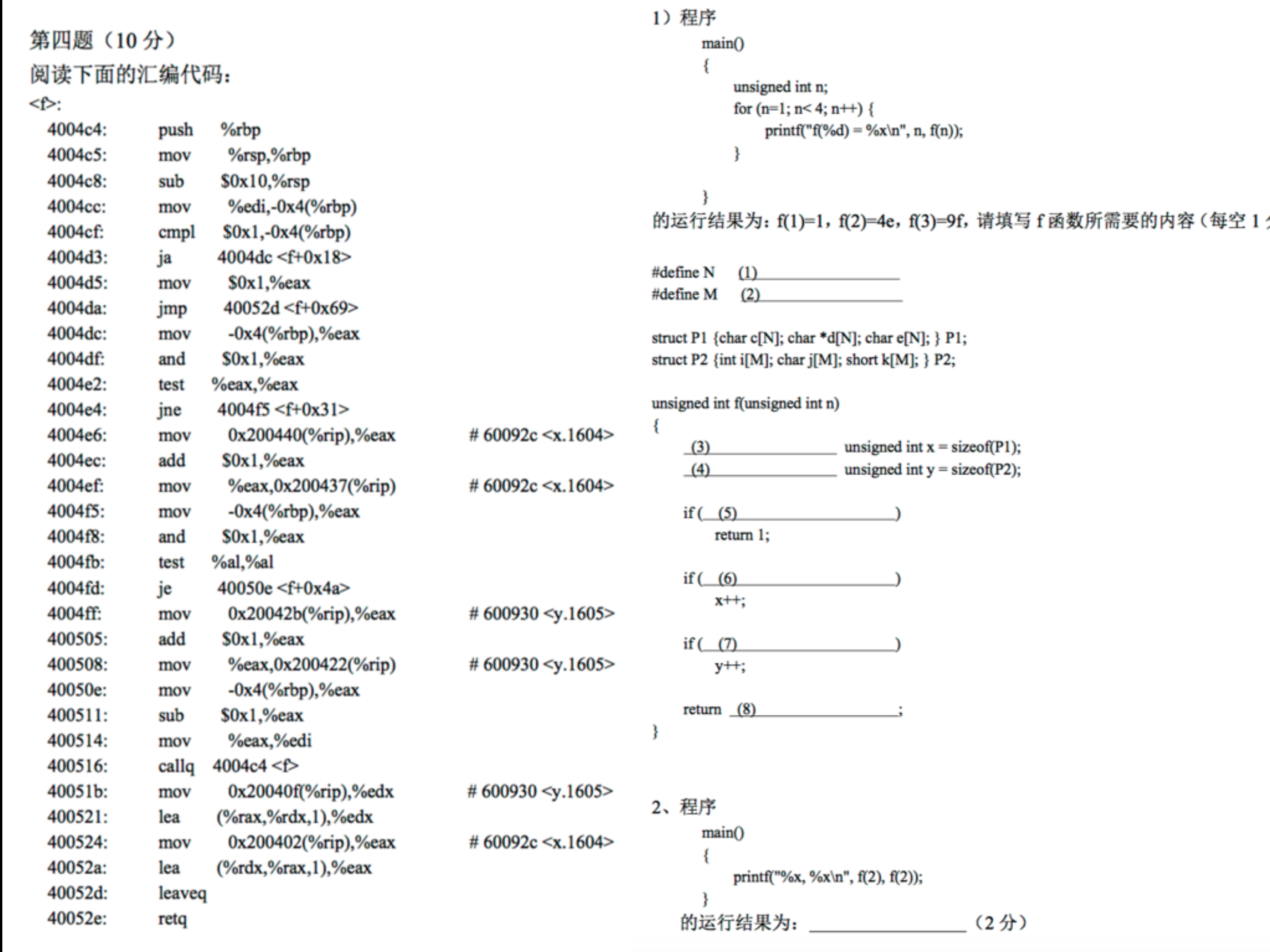
- 3. Alice is having trouble with GDB. She asks Bob how to set a breakpoint, and he shows her.
  - Who cheated: Bob? Alice?
- 4. Charlie goes to a TA and asks for help
  - Who cheated: Charlie?
- If you are uncertain which of these constitutes cheating, and which do not, please read the syllabus carefully. If you're still uncertain, ask one of the staff

HOW TO LEARN *ICS*

- Four level
  - I. PPT — Essential
  - II. Textbook — Read after class
  - III. Practice — Program in Linux
  - IV. Lab — Challenge yourself !

HOW TO LEARN *ICS*

- Exam
  - Much more complex and difficult than examples and exercises
  - **Proficiency**



第四题（10 分）

阅读下面的汇编代码：

```
<f>:
  4004c4:     push    %rbp
  4004c5:     mov      %rsp,%rbp
  4004c8:     sub     $0x10,%rsp
  4004cc:     mov      %edi,-0x4(%rbp)
  4004cf:     cmpl    $0x1,-0x4(%rbp)
  4004d3:     ja      4004dc <f+0x18>
  4004d5:     mov      $0x1,%eax
  4004da:     jmp     40052d <f+0x69>
  4004dc:     mov      -0x4(%rbp),%eax
  4004df:     and     $0x1,%eax
  4004e2:     test    %eax,%eax
  4004e4:     jne     4004f5 <f+0x31>
  4004e6:     mov      0x200440(%rip),%eax          # 60092c <x.1604>
  4004ec:     add     $0x1,%eax
  4004ef:     mov      %eax,0x200437(%rip)          # 60092c <x.1604>
  4004f5:     mov      -0x4(%rbp),%eax
  4004f8:     and     $0x1,%eax
  4004fb:     test    %al,%al
  4004fd:     je      40050e <f+0x4a>
  4004ff:     mov      0x20042b(%rip),%eax          # 600930 <y.1605>
  400505:     add     $0x1,%eax
  400508:     mov      %eax,0x200422(%rip)          # 600930 <y.1605>
  40050e:     mov      -0x4(%rbp),%eax
  400511:     sub     $0x1,%eax
  400514:     mov      %eax,%edi
  400516:     callq   4004c4 <f>
  40051b:     mov      0x20040f(%rip),%edx          # 600930 <y.1605>
  400521:     lea     (%rax,%rdx,1),%edx
  400524:     mov      0x200402(%rip),%eax          # 60092c <x.1604>
  40052a:     lea     (%rdx,%rax,1),%eax
  40052d:     leaveq
  40052e:     retq
```

1）程序

```
main()
{
    unsigned int n;
    for (n=1; n< 4; n++) {
        printf("f(%d) = %x\n", n, f(n));
    }

}
```

的运行结果为：f(1)=1，f(2)=4e，f(3)=9f，请填写 f 函数所需要的内容（每空 1

```
#define N    (1)_______________
#define M    (2)_______________

struct P1 {char c[N]; char *d[N]; char e[N]; } P1;
struct P2 {int i[M]; char j[M]; short k[M]; } P2;

unsigned int f(unsigned int n)
{
    (3)_______________ unsigned int x = sizeof(P1);
    (4)_______________ unsigned int y = sizeof(P2);

    if ( (5)_______________ )
        return 1;

    if ( (6)_______________ )
        x++;

    if ( (7)_______________ )
        y++;

    return (8)_______________;
}
```

2、程序

```
main()
{
    printf("%x, %x\n", f(2), f(2));
}
```

的运行结果为：_______________（2 分）

TUTORIALS

- Every Thursday 18:30 – 20:30
- Room 307, Natural Science Teaching Building
  - 1st period: Review of lectures
  - 2nd period: Review of labs / homework / quiz

LET’S GET STARTED!

●アニメディア 2014年1月号初出
原画／石原恵治　監修／近岡直　仕上げ／J.C.STAFF
背景／江口祥重　撮影／木村実乃理(Studio Shamrock)
©Green Leaves / Wake Up, Girls! 製作委員会