CS490: Problem Solving in Computer Science

Lecture 2: FAQ and Common Programming Datatypes

Dustin Tseng
Mike Li

Fridayday January 6, 2006

Outline

- Registration
- Frequently Asked Questions
- Programming Primitive Datatypes

- Registration
- Frequently Asked Questions
- Programming Primitive Datatypes

People Who I Will Register

- Hans Lee
- Ke Kay Miao
- Faisal Al-Humoud
- Luqman Masood
- Brian Luus
- Sandy Robertson
- Andrew Juren
- Clement Keng-Li Lim

People That Aren't Sure

- James McRoberts
- Mark Jia
- Sharareh Sherry Farzaneh
- Peter Wong

People That Want To Leave?

- Clement Keng-Li Lim
- Henry Ho
- Philip Hsu Hua Chu Hsu

- Registration
- Frequently Asked Questions
- Programming Primitive Datatypes

FAQ

Q: Can I still register?

A: Yes, but NOT FOR LONG!
Talk to me TODAY!

FAQ

Q: Is this an easy class?

A: No. It is a forth year course.

FAQ

Q: How much work load is it then?

A: It should be around the workload of CPSC 320/420, or any other high level CPSC courses. During the week of your group's presentation, expect more work (in return you get no final!)

FAQ

Q: How do I find group members? How many do I need?

A: Each group should be 2-3 people in size.
We're planning to leave sometime today for students to meet each other. Why don't we have an intro from everyone now?

FAQ

Q: When do I have to pick a topic?

A: ASAP, please have some topic in mind on Monday.
By next Wednesday, everyone should have a topic, unless there is a conflict.
By next Friday, we're planning to have the topics all assigned.

FAQ

Q: How will I ever start a presentation?

A: There's several places to start. Textbooks such as Goodrich and Cormen can help you with theoretical parts (and a little bit of coding). If possible, coordinators will start you off with some stuff from last year.

FAQ

Q: How will I be marked on presentations?

A: Right now we have peer review in mind. That includes the coordinators :\)

FAQ

Q: Where can I see more examples of problems similar to that of the course?

A: UVA `http://acm.uva.es/problemset/`
TopCoder `http://www.topcoder.com/tc`
These are also good problem sources when you're presenting.
Igor's UVA tools `http://shygypsy.com/acm/`
Demo later if we have time.

FAQ

Q: What should I be doing now?

A: You should be listening to this magnificient speech. After class don't forget to:

- talk to us about registration (if needed)
- look for group partners
- keep thinking about the topics

- Registration
- Frequently Asked Questions
- Programming Primitive Datatypes

Integers in C/C++

`int` :

- size: 32-bit, 4 bytes
- range: [-2,147,483,647, 2,147,483,647]

`long` :

- size: 32-bit, 4 bytes
- range: [-2,147,483,647, 2,147,483,647]

`long long` :

- size: 64-bit, 8 bytes
- range: [-9,223,372,036,854,775,808, 9,223,372,036,854,775,807]

Integers in C/C++

`unsigned int` :

- size: 32-bit, 4 bytes
- range: [0, 4,294,967,295]

`unsigned long` :

- size: 32-bit, 4 bytes
- range: [0, 4,294,967,295]

`unsigned long long` :

- size: 64-bit, 8 bytes
- range: [0, 18,446,744,073,709,551,615]

Integers in C/C++

`int` :

- size: 32-bit, 4 bytes
- range: [-2,147,483,647, 2,147,483,647]

`long` :

- size: 32-bit, 4 bytes
- range: [-2,147,483,647, 2,147,483,647]

`long long` :

- size: 64-bit, 8 bytes
- range: [-9,223,372,036,854,775,808, 9,223,372,036,854,775,807]

Integers in Java

`int` :

- size: 32-bit, 4 bytes
- range: [-2,147,483,647, 2,147,483,647]

`long` :

- size: 64-bit, 8 bytes
- range: [-9,223,372,036,854,775,808, 9,223,372,036,854,775,807]

`BigInteger` :

- a class defined in Java API
- handles arbitrarily large integers
- BigInteger(String val)
- BigInteger add(BigInteger val)
- see API for more detail

Floating Points in C/C++ and Java

`float` :

- size: 32-bit, 4 bytes
- range: 3.4E +/- 38
- precision: 7 digits

`double` :

- size: 32-bit, 8 bytes
- range: 1.7E +/- 308
- precision: 15 digits

`long double` (C/C++ only) :

- size: 80-bit, 10 bytes
- range: 1.2E +/- 4932
- precision: 19 digits

Comparing floating points

Using the comparison operator is a bad idea, since sometimes two numbers are effectively equal.
Instead, we usually do the following:

```
double eps = 1e-7;

function equal(double a, double b) {
// instead of using (a==b)
// where a and b are compared bit by bit
return (abs(a-b) < eps)
}
```

What Else?

- Demo on UVA: `http://acm.uva.es/p/v100/10055.html`
- Form groups if havent done that
- Think about topics
- Feedbacks (even on the evaluation form itself)