TA Induction Workshop Spring 2010

Effective Teaching Skills

Dr Otis Lam

My Post-TA experience

Organization
Teaching
Instruction
Support to Students

Learning
Assessment – e.g. Grading
Management

Your expectation?

What is the biggest concern about TA in your mind now?

,or

What do you want to take away from this workshop?

Overview

- **Being a TA**

    Roles & Responsibilities
    Good vs Bad TAs

- **Common Problems** and ways to handle them

- **Teaching resources**

Being a TA

Roles and responsibilities

Change of Role

Graduate

Teaching Assistant

What am I supposed to do??
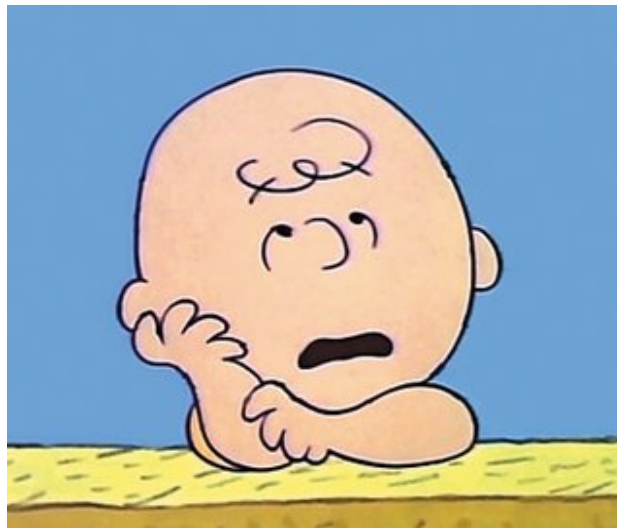
GAP

HKUST said…

💧 Roles and Duties:
http://celt.ust.hk/ta/tahome02.htm

"Teaching assistants, a group of active postgraduate students **who bridge the gap between the instructors and the students.** …"

No matter how competent you're, It's pointless if you don't know where is the gap (responsibilities).

How much do you know about it?

Give me some examples.

Possible responsibilities of a TA

Conduct Tutorials

Conduct labs

Mark quizzes, tests homework assignment

Conduct review sessions

Responsibilities

Create solution Sets

Meet with students

Invigilate Examinations

Setup a class email, list server or website

Attend Pre-Lab

Become familiar with lecture assignments

Attend lectures
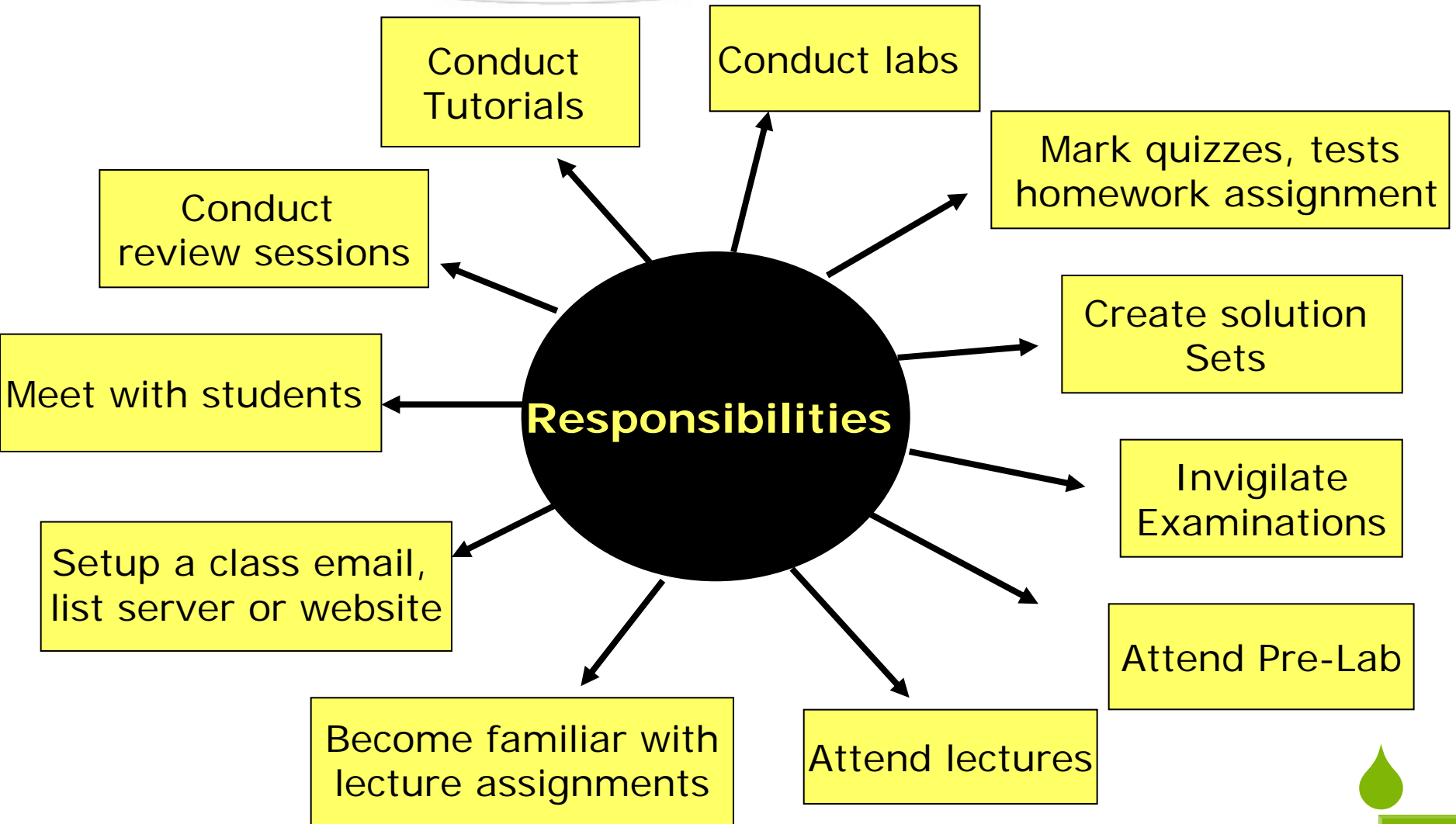
Clarify with the right person at a right time!

Prepare BEFORE you start

1) Clarify your responsibilities at the beginning

- **Course**

- **TAC**

- **The other TAs (current/previous)**

2) TAKE INITIATIVE for your own sake. (They may or may not approach you.)

© Center for Enhanced Learning and Teaching, HKUST

What questions should I ask my instructor/professor?

A quick checklist

Handout:
"Questions To Ask The Instructor"

This is to help you:
- **To think about the right questions to ask**
- **To check if there is anything unclear or missing about your duties**

© Center for Enhanced Learning and Teaching,

We are here to support you!

Center for Enhanced Learning and Teaching (CELT) 教學促進中心 &

The Teaching Assistant Coordinator (TAC)

Being a TA

Roles and responsibilities

Good vs Bad TAs

Group Discussion
Good vs Bad TA

- **Discuss in groups of 3 and come up with Three characteristics of a good TA and a bad TA.**

From your past experience:

- **What do you think a good TA is like?**

- **What do you think a bad TA is like?**

~ 5 Minutes ~

Characteristics of a **good** TA

Summary

- Time management between being a TA and PG student

- Good preparation before class

- Good relationship with students (charming, friendly, out-going

- fair in marking and grading

- Good communication/interpersonal skills

- Teamwork skills

Characteristics of a **bad** TA

Summary

- **Start late, end late**

Being a TA

Roles and responsibilities

Good vs Bad TAs

Common Problems of TAs

Common problems of TA

Insufficient preparation

Nervous

Too fast

Too much information

Boring

Nervousness

In particular the first lesson…

Nervousness

It's just **normal**. You will get over.

Some tips for the 1ˢᵗ day

- Check out the place before the 1ˢᵗ lesson
- Arrive earlier
- Chat with the students who arrive early
- Clean up the white board (every time)
- Write down your name and course name on the board
- Explain the assessment and the course policy clearly
- DON'T distribute the syllabus and dismiss the class

Preparation! Preparation! Preparation!

1. Course content, administration and policy
(read the course outline)

2. Do the pre-lab properly AND attend related lectures

3. Read supplementary reading

© Center for Enhanced Learning and Teaching, HKUST
http://www.ahajokes.com/cartoon/readdum.jpg

Insufficient Preparation

I don't have enough time…

Time Management Workshop

TA 124

Balancing Time between
TA Duties and Research

By Dr Lucia Yeung

Common problems of TA

Insufficient preparation

Nervous

Too fast

Too much information

Boring

Too Fast

Presentation Skills

Too much information

http://wwwdelivery.superstock.com/WI/223/1538/PreviewComp/SuperStock_1538R-49866.jpg

Presentation Skills

- **Speak slower**

- **Use powerpoint / Notes**

- **Ask students if they have any questions**

- TA113 Effective Presentation Skills

- This is a 1.5-hour workshop <u>mandatory</u> for ALL new TAs to attend. It aims to (1) introduce TAs strategies for effective presentation, and (2) conduct group activities and case studies to consolidate what TAs have just learnt and to let them review their own presentations. After the workshop, TAs will learn how to prepare, construct and deliver successful presentations during their teaching.

- TA 131 Advanced PowerPoint Presentation

PHP Skinny Sheet

VisiBone

```
array (
  'seconds' => 59.
  'minutes' => 59.
  'hours' => 23.
  'mday' => 31.
  'wday' => 5.
  'mon' => 12.
  'year' => 1999.
  'yday' => 364.
  'weekday' => 'Friday'.
  'month' => 'December'.
  0 => 946702799
```

```php
class Voter extends Citizen {
    var $party;
    function Voter($name) {
        parent::Citizen($name);
    }
    function register($party) {
        $this->party = $party;
    }
}
$o = new Voter('John Q.');
```

```php
interface Lawyer {
    public function argue(array $cases);
}
class Senator extends Voter implements Lawyer {
    static private $record = '';
    final public function argue(array $cases) {
        Senator::$record .= join($cases);
    }
    protected function influence(Lobbyist $x) {}
}
```

Function Tools

```php
$f = create_function('$x', 'return 2*$x;');  assert(4 == $f(2));
function f() { return func_get_args(); } assert(array(1,'two') = f(1,'two'));
```

Operator Precedence

```php
$o = new ClassName;
$o = new $sClassName;

$Z = $a[$i];  // array element by numeric index
$Z = $a[$s];  // array element by string key
$s = $s[$i];  // a character from a string
```

```php
$n++;  ++$n;  // post- and pre-increment
$n--;  --$n;  // post- and pre-decrement

$i = ~$i;  // bitwise complement
$n = -$n;
$Z = @$Z;  // hide error messages
$b = (bool)$z;  $b = (boolean)$z;
$i = (int)$z;  $i = (integer)$z;
$x = (double)$z;  $x = (float)$z;  $x = (real)$z;
$s = (string)$z;
$a = (array)$z;
$o = (object)$z;
NULL == = (unset)$Z;
```

```php
$b = $Z instanceof C;
$b = $Z instanceof $sClassName;
$b = $Z instanceof $o;

$b = !$b;  // boolean not
```

```php
$n = $n * $n;
$n = $n / $n;
$i = $i % $i;  // integer modulo

$a = $a + $a;  // union by keys, not values
$n = $n + $n;
$n = $n - $n;
$s = $s . $s;  // string concatenate, à la Perl

$i = $i << $i;  // shift bits left
$i = $i >> $i;  // shift bits right

$z = $z < $z;  // less
$z = $z <= $z;  // less or equal
$z = $z >= $z;  // more or equal
$z = $z > $z;  // more

$b = $Z == $Z;  // loose equal      not = assignment
$b = $Z != $Z;  // not loosely equal
$b = $Z <> $Z;  // inequal
$b = $Z === $Z;  // strictly equal (same type)
$b = $Z !== $Z;  // not strictly equal

$i = ($i & $i);  // bitwise 'and'    use parens
$Z = &$Z;  // assign (or pass or return) by reference

$i = ($i ^ $i);  // bitwise 'xor'    use parens

$i = ($i | $i);  // bitwise 'or'    use parens

$b = $b && $b;  // boolean logic  = $b and $b

$b = $b || $b;  // boolean logic  = $b or $b

$Z = ($b ? $Z : $Z);  // if-else chooser    use parens

$Z = $Z;  // assign by value (copy)
$n += $n;  // add to, etc.
$n -= $n;
$n *= $n;
$n /= $n;
$i %= $i;
$s .= $s;  // concatenate to
$i &= $i;
$i |= $i;
$i ^= $i;
$i <<= $i;
$i >>= $i;

$b = $b and $b;  // boolean logic  = $b && $b

$b = $b xor $b;  // boolean logic

$b = $b or $b;  // boolean logic  = $b || $b
```

Specify the class name, a string containing the class name, or an object instance of the class.

Is $Z an object of the class? Or a class that extends or implements the class?

PHP types

gettype()

$x "double" aka float, real

$n numeric

Array Simple

```php
$a = array(2, 'b', 'or' =>'not', array('2b'), );  // trailing comma ok
assert(4 == count($a)
    && 2 == $a[0]
    && 'b' == $a[1]
    && 'not' == $a['or']
    && '2b' == $a[2][0]);
assert(3 == count(array(7,8,9));
assert(6 == count(array(7,8,array(9,10,11)), COUNT_RECURSIVE));
assert(array('I', 'am') = explode(' ', 'I am'));
assert('I am' == implode(' ', array('I', 'am')));  //  = join()
assert(TRUE == in_array(333, array(1, 2, '333', 4)));
assert('I am' == join(' ', array('I', 'am')));
list($one,$two) = array(1,2);  assert(1 == $one && 2 == $two);

assert(array(1,2,3,4,5) == range(1,5));
$a = array(1,2,3,...
```

Simple Math

```php
assert(5 == abs(-5));
assert(2.0 == ceil(1.1) && -1.0 == ceil(-1.9));
assert(2.0 == floor(2.9) && -3.0 == floor(-2.1));
assert(2.5 == fmod(12.5, 10.0));
assert(FALSE == is_finite(9e9999));
assert(TRUE == is_infinite(9e9999));
assert(0.0 <= lcg_value() && lcg_value() <= 1.0);  // random
assert(9 == max(7,8,9) && 'y' == max('x','y','Z'));  // CScs
assert(1 == min(3,2,1) && 'a' == min('c','b','a'));
assert(0 <= mt_rand() && mt_rand() <= mt_getrandmax());
    $i = mt_rand(-1000,1000);  assert(abs($i) <= 1000);
    mt_srand(0);  assert(963932192 == mt_rand());
assert(0 <= rand() && rand() <= getrandmax());
assert(-100 <= rand(-100,100) && rand(-100,100) <= 100);
assert(-2.67 == round(-2.666, 2));
srand(0);  assert(12345 == rand());
```

Array Sort

```php
assert(array_multisort($a,$b) && array(1,1,2) == $a && array(8,9,0) == $b);
assert(arsort($a) && array(2 =>9, 1 =>8, 0 =>7) == $a);
assert(asort($a) && array(2 =>7, 1 =>8, 0 =>9) == $a);
assert(krsort($a) && array(9 =>1, 8 =>1, 7 =>1) == $a);
assert(ksort($a) && array(7 =>1, 8 =>1, 9 =>1) == $a);
assert(natcasesort($a) && array(1 =>'a99', 0 =>'A222') == $a);
assert(natsort($a) && array(1 =>'a99', 0 =>'a222') == $a);
assert(rsort($a) && array(9,8,7) == $a);
assert(sort($a) && array(7,8,9) == $a);
assert(uasort($a, 'strnatcasecmp') && array(1 =>'k9', 0 =>'K12') == $a);
assert(uksort($a, 'strnatcmp') && array('b4' =>0,'b12' =>0) == $a);
    $a = array('c','a','B');  assert(uksort($a, create_function('$L,$R', 'return strcasecmp($L,$R);'))
        && array('a','B','c') == $a);
```

Type Test

```php
guess('stdClass' == get_class($o));
assert('mysql link' == get_resource_type($link));
assert('integer' == gettype(9));
    assert(is_array(array(1,2,3)));
    assert(is_bool(FALSE));
    assert(is_callable('strlen'));
    assert(is_double(7.5));
    assert(is_float(7.5));
    assert(is_integer(3) && is_int(3));
    assert(is_null(NULL));
    assert(is_numeric(2.55e2));
    assert(is_object(new stdClass));
    assert(is_real(7.5));
    guess(is_resource(mysql_connect('x.com')));
    assert(is_scalar('s') && !is_scalar(array()));
    assert(is_string('abc'));
```

Variables are function-local. The two ways to use a global variable within a function:
global $variable; // once at function top
$GLOBALS['variable'] // anywhere.

Every function behaves as if it had these automatically:
global $GLOBALS, $_COOKIE, $_ENV, $_FILES, $_GET;
global $_POST, $_REQUEST, $_SERVER, $_SESSION;

Forms

```php
guess('file.jpg' == $_FILES['pic']['name']);  // e.g. from <input type='file' name='pic'>
guess('value' == $_GET['field']);  // aka $HTTP_GET_VARS[] e.g. <input name=\'field\'>
guess('value' == $_POST['field']);  // aka $HTTP_POST_VARS[]
guess('value' == $_REQUEST['field']);  // merged $_GET[] and $_POST[] etc.
```

Output

```php
?>odd-looking output command<?php
echo("with ".parentheses");
echo "without ", "parentheses";
print "almost identical to echo()";
3.14 assert(4 == printf('%4.2f', M_PI));
assert("Array\n(\n    [0] => 33\n)\n" == print_r(array(33), TRUE));
assert("array (\n 0 => 4,\n 1 => 'ensic',\n)" == var_export(array(4,'ensic'), TRUE));
3.14 assert(4 == vprintf('%4.2f', array(M_PI)));
```

```php
print_r(array(4, 'ensic'));
```
```
Array
(
    [0] => 4
    [1] => ensic
)
```

```php
var_dump(array(4, 'ensic'));
```
```
array(2) {
  [0]=>
  int(4)
  [1]=>
  string(5) "ensic"
}
```

```php
var_export(array(4, 'ensic'));
```
```
array (
  0 => 4,
  1 => 'ensic',
)
```
(outputs PHP syntax)

```php
$to = "\"Reese E. Vuur\" <to@x.com>";
$subject = "Unfiltered user input is dangerous";
$headers = "From: \"S.N.Dürr\" <fm@x.com> \r\n";
$message = "Multi-\nline\nmessage.\n";
assert(mail($to, $subject, $message, $headers));
```

Toolbox

```php
assert(4 == 2 + 2);
assert(4 == 2 + 2');  // more informative failure
define('THREE', 3);  assert(3 == THREE);
define('TWO', 2);  assert(defined('TWO'));  // remember quotes
hopethisreturnstrue() or die('dashed hope');
$z = array();  assert(empty($z));  $z = '';  assert(empty($z));
guess('linux' == $_ENV['TERM']);
if (errorlevel99()) { exit(99); } else { exit('message'); }
$a = getallheaders();  guess(preg_match('/Mozilla/', $a['User-Agent']));
global $z;  $z = 'one way';  // (inside a
$GLOBALS['z'] = 'another way';  // function)
include 'localfile';
include_once('localfile.php');
$q = 2;  assert(isset($q) && !isset($undefinedvariable));

ob_start();  echo 'output';  $s = ob_get_clean();  assert('output' == $s);
phpinfo();  // password-protect this report
guess('5.2.5' == phpversion());
require 'localfile.php';
require_once 'localfile.php';
guess('/home/username/htdocs' == $_SERVER['DOCUMENT_ROOT']);
guess('Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)' == $_SERVER['HTTP_USER_AGENT']);
guess('http://host.com/frompage.html' == $_SERVER['HTTP_REFERER']);  // (sic)
guess('/dir/file.php' == $_SERVER['PHP_SELF']);
guess('217.5.153.85' == $_SERVER['REMOTE_ADDR']);
guess('/home/username/path/file.php' == $_SERVER['SCRIPT_FILENAME']);
$q = 2;  assert(!isset($q));  unset($q);  assert(!isset($q));
```

Time & Date

```php
assert('23:59:59 Fri 31-Dec-1999' == date('H:i:s D d-M-Y',mktime(23,59,59,12,31,1999)));
assert('00:00:00 Sat 01-Jan-2000' == date('H:i:s D d-M-Y',mktime(23,59,59,12,31,1999) + 1));
var_export(getdate(mktime(23,59,59,12,31,1999)));  // →
    $a = getdate(time());  guess('Monday' == $a['weekday']);
guess('0.20934800 1200089159' == microtime());
    guess(1200089159.209348 == microtime(TRUE));
    assert(1 == mktime(0,0,0,1,1,2000)-mktime(23,59,59,12,31,1999));  // y2k moment
    $t1 = time();  dosomething();  $t2 = time();  $nSeconds = $t2-$t1;
```

Integer

```php
assert(44 == (int)...
assert(44 == = (int...
assert(44 == = 0+'...
assert(2.0 == double...
assert(-2.0 == floatva...
assert(-3 == intval(...
```

String

```php
assert('44' == = (string...
assert('44' == = '' . 44);
assert('3.14' == = strval(...
```

© 2008 VisiBone   PHP Skinny Sheet   April 2008 Edition   PHP 4,5   www.visibone.com/php

Too much information

- 7 ±2 rule

- **Selection**
  - Primary focus
  - Things that students don't understand the most

- **Coherent**

Common problems of TA

Insufficient preparation

Nervous

Too fast

Too much information

Boring

Too Boring

Question
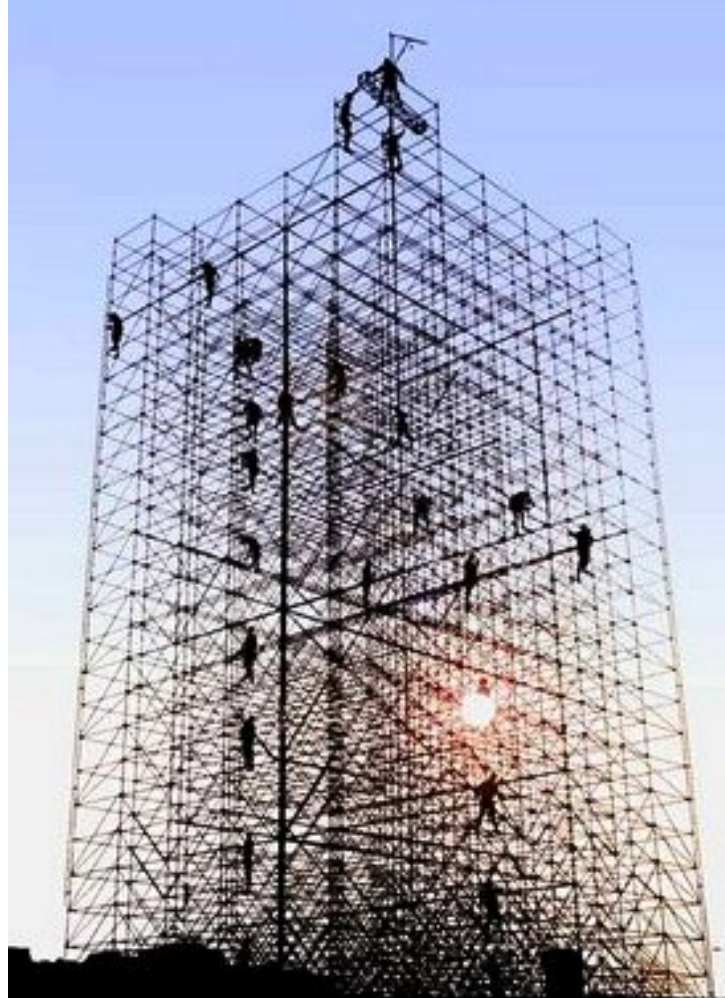
Why would you feel <u>bored</u> in a class?

➢

➢

➢

➢

Too Boring...

- Too difficult/ Too simple

- Lack of motivation

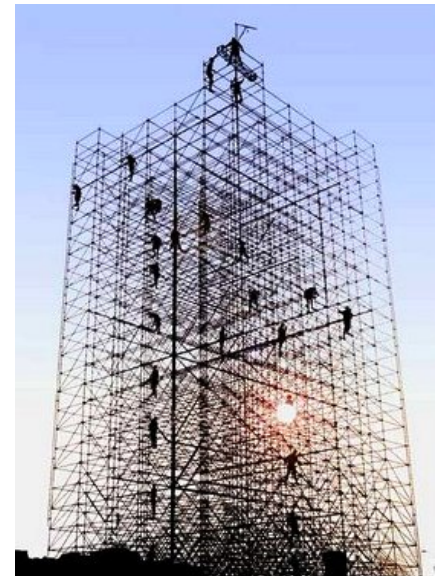- Disengagement
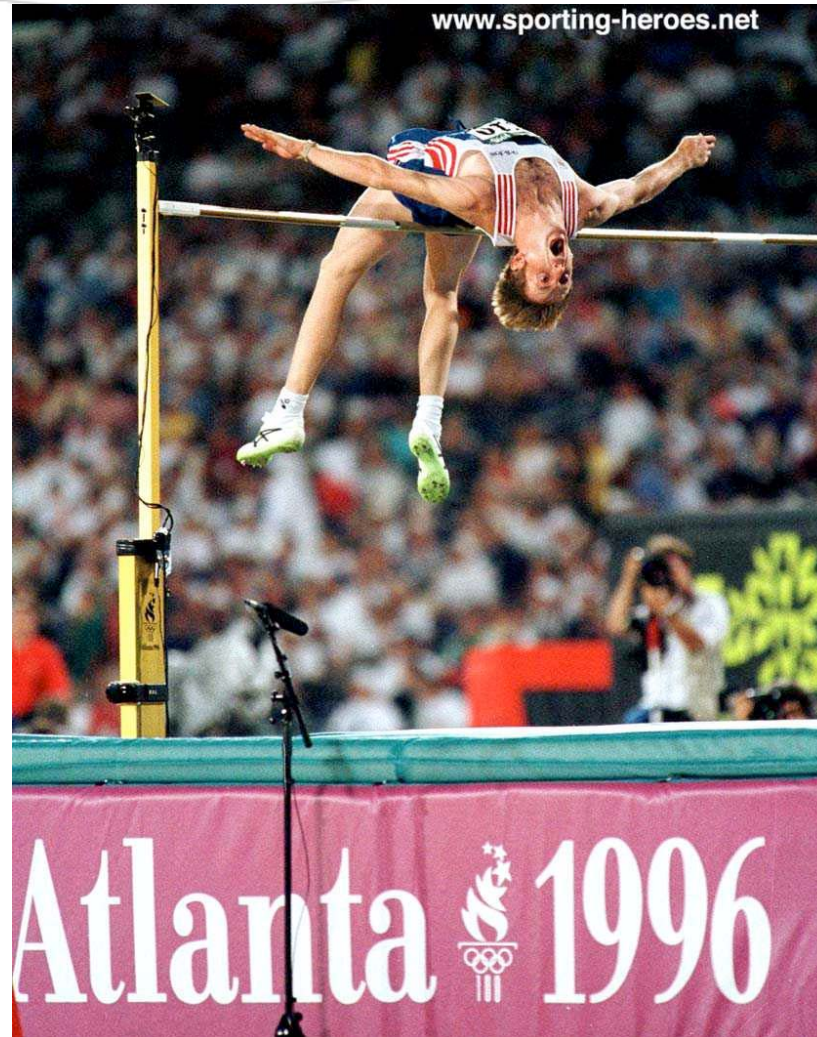 - e.g. Unfairness, bias
  - e.g. Presentation

Scaffolding

http://www.serendipity35.net/uploads/scaffolding.jpg

Scaffolding

- Built/ created by students themselves

- New knowledge is built upon prior knowledge

Teach at the right level

www.sporting-heroes.net

Atlanta 🔥 1996

Scaffolding

- Assess the current level of your students (e.g. by asking questions)

- Decide the next level of appropriate instruction

- Assess if students have moved to the next level
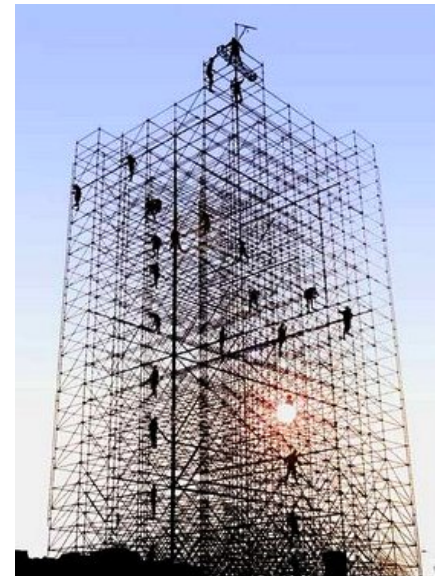
Scaffolding

EXAMPLE:

- **Current Level:** 2 digit addition

$$25+81=?$$

- **Next Level:** ??

- **How to assess?**
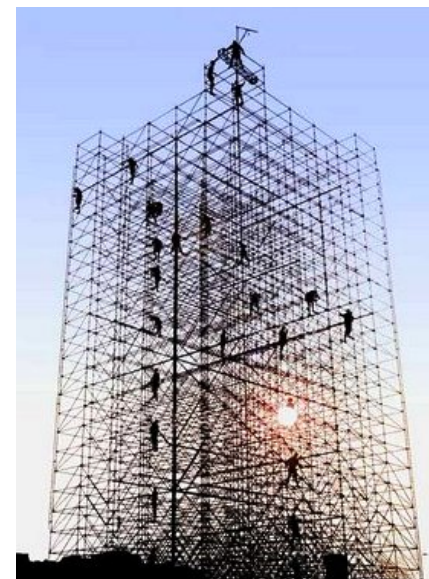
Too Boring...

- Too difficult/ Too simple

- Lack of motivation

- Disengagement
  - e.g. Unfairness, bias
  - e.g. Presentation

Problem with Motivation

🔹 Many students just focus on **personal interests** of academic tasks. If they are not interested, then they are not motivated.

Problem with VALUE

- "I'm not interested in this…"

- "I study Chemistry, I don't see any value of this major in my future career…so I'm not motivated to learn all these equations…"

- **Question:** What is the value of studying for a major that is NOT related to students' future work?

The Role of Task Value Beliefs

- **Intrinsic Value** - Enjoyment

- **Attainment Value** – Importance of doing well on a task
  e.g. student and self-identity

- **Utility Value** – Usefulness of tasks for future goals.
  e.g. organic chemistry and doctor

Reference: Eccles, J. & Wigfield, A. (1995). In the mind of the actor: The structure of adolescents' achievement task values and expectancy-related beliefs. *Personality and Social Psychology Bulletin*, 21, 215-225.

Too Boring...

- Too difficult/ Too simple
- Lack of motivation
- Disengagement
  - e.g. Unfairness, bias
  - e.g. Presentation

Common problems of TA

Insufficient preparation

Nervous

Too fast

Too much information

Boring

Summary

- Being a TA Roles and responsibilities

  Good vs Bad TAs

- Common Problems of TAs and ways to handle them

- Teaching resources

⬥ **Any Questions?**

Available Teaching Resources

Available resources

GOOD LEARNING ENVIRONMENT

At the start of the semester, you may wish to highlight to your students your expectation for academic integrity and good classroom behavior. To support you on this, we have produced a short PPT presentation (http://www.ust.hk/vpaao/conduct/good_learning_experience.pps) for your reference. Feel free to adopt this for your use, or to edit this resource as you wish.

CONDUCT IN THE CLASS ROOM

Behaviors that will facilitate/disrupt learning in the classroom
http://www.ust.hk/vpaao/conduct/classroom_conduct.ppt

Available resources

LARGE CLASS TEACHING

http://celt.ust.hk/experience/largeclass/index.html
Maintaining a good learning atmosphere is more difficult for large classes, and hope some of these ideas on teaching large classes and fostering active learning are useful to you.

**A GUIDE TO GOOD REFERENCING SKILLS
(to avoid plagiarism)**

http://lcms01.ust.hk/sbm/wsc/referencing/introduction/index.html

ACADEMIC INTEGRITY

http://www.ust.hk/vpaao/integrity
UST places a strong emphasis on academic integrity and has introduced regulations to back this up. To help students and staff to understand the policy, a website has been established that explains the regulations, provides assistance for students in avoiding plagiarism, and sets out the role of faculty and staff when a case of cheating or plagiarism comes to their attention.

"Do to others as you would have them do to you."

Luke 6:31

The End

Thank You