TA Induction Workshop Fall 2009

Effective Teaching Skills

Dr Lucia Yeung / Dr Otis Lam

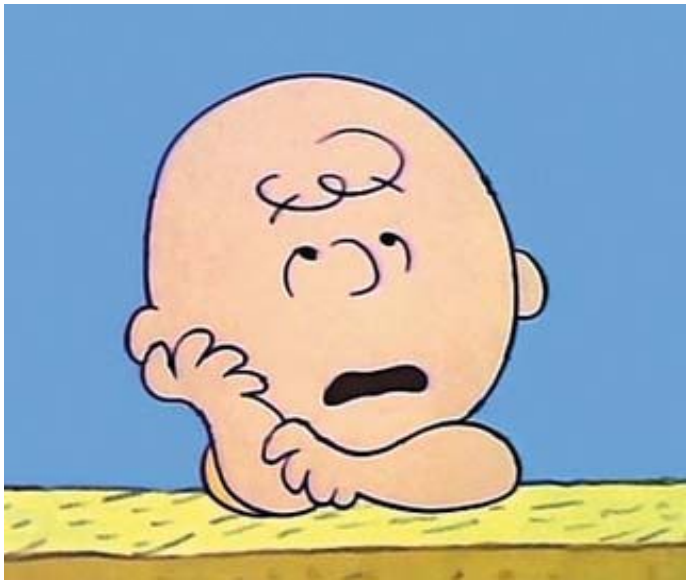
Change of Role

Graduate

Teaching Assistant

http://www.molecularstation.com/molecular-biology-images/data/507/cartoonGrad.gif
http://static.guim.co.uk/Guardian/culture/gallery/2008/oct/23/television/GD9313478@Title-BOY-NAMED-CHARL-8397.jpg

What am I supposed to do??

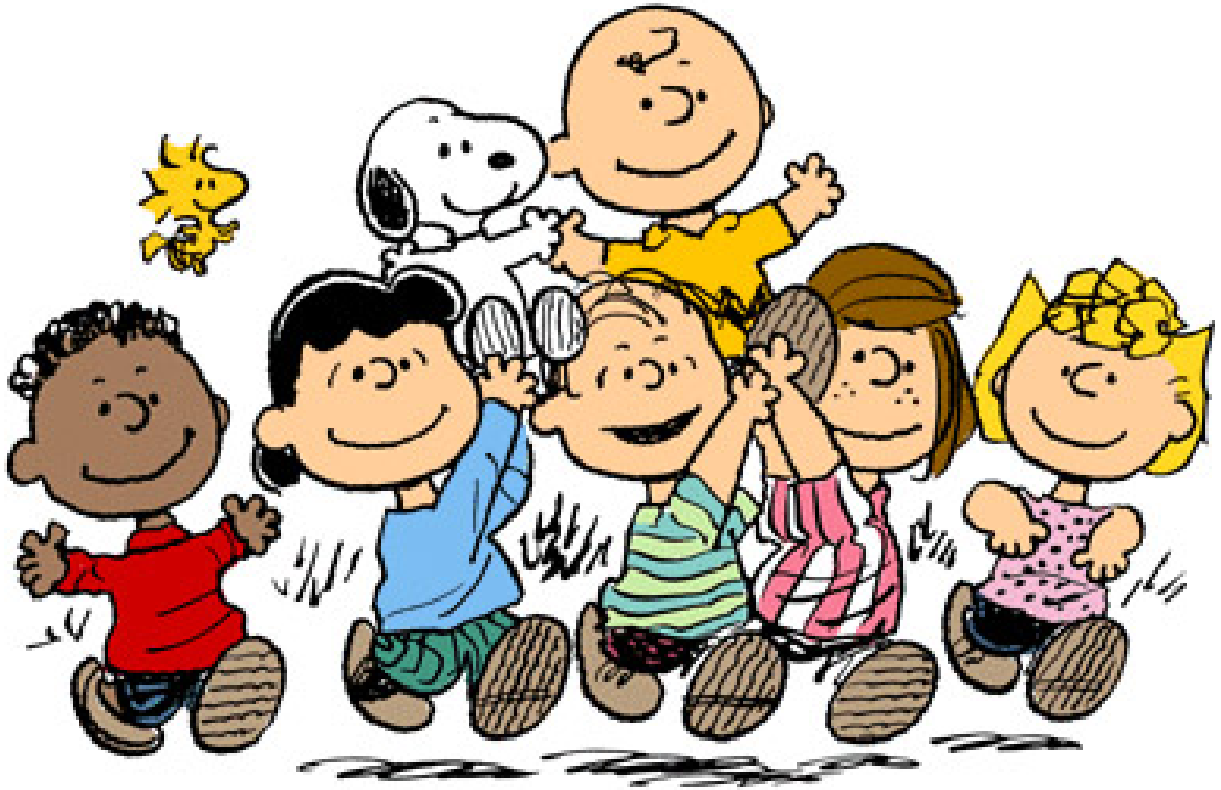


http://2.bp.blogspot.com/_V9mssXHqYmo/RgPeVDU4sOI/AAAAAAAAADM/763IASPtKWg/s320/charlie%2Bbrown.jpg

We are here to support you!

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

The Teaching Assistant Coordinator (TAC)



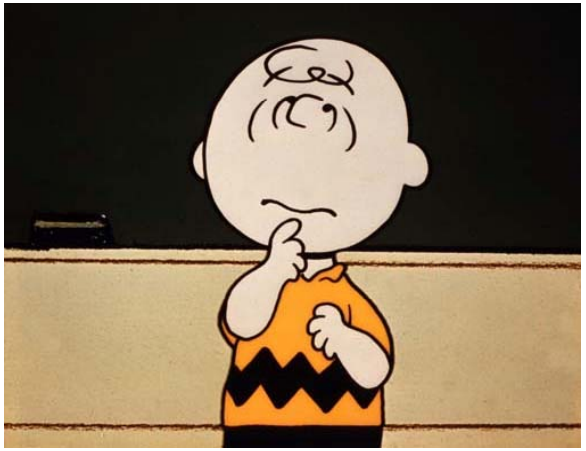
http://www.thepurplemoon.com/blog/peanutsgang.jpg

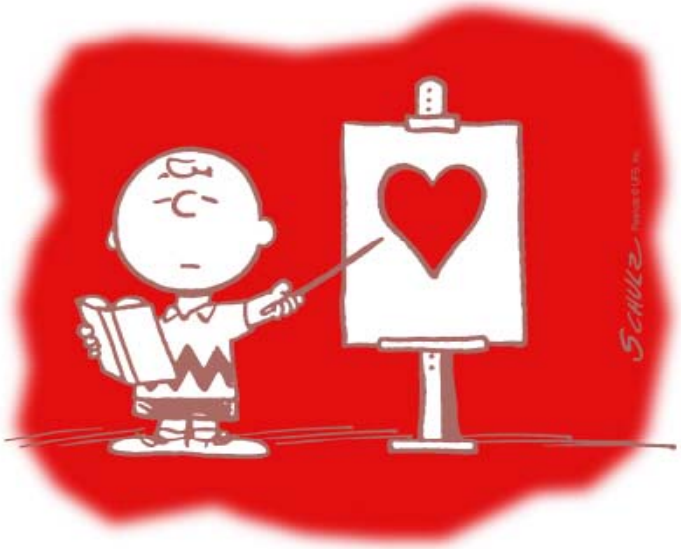
Intended Learning Outcomes

- By the end of this workshop, you should be able to:
- **1.** Describe the **roles and duties** of being Teaching Assistant at UST
- **2.** Explain **Common problems in teaching** and ways to handle them.
- **3.** Identify **teaching resources** available to support you

ROLES & DUTIES

Teaching Assistant





http://images.cafepress.com/image/17394441_400x400.jpg
http://static.guim.co.uk/Guardian/culture/gallery/2008/oct/23/television/GD9313478@Title-BOY-NAMED-CHARL-8397.jpg
http://images.clipartof.com/small/17702-Clipart-Illustration-Of-An-Orange-Man-Scientist-Holding-A-Test-Tube-Full-Of-Bubbly-Orange-Liquid-In-A-Laboratory.jpg

ROLES & DUTIES

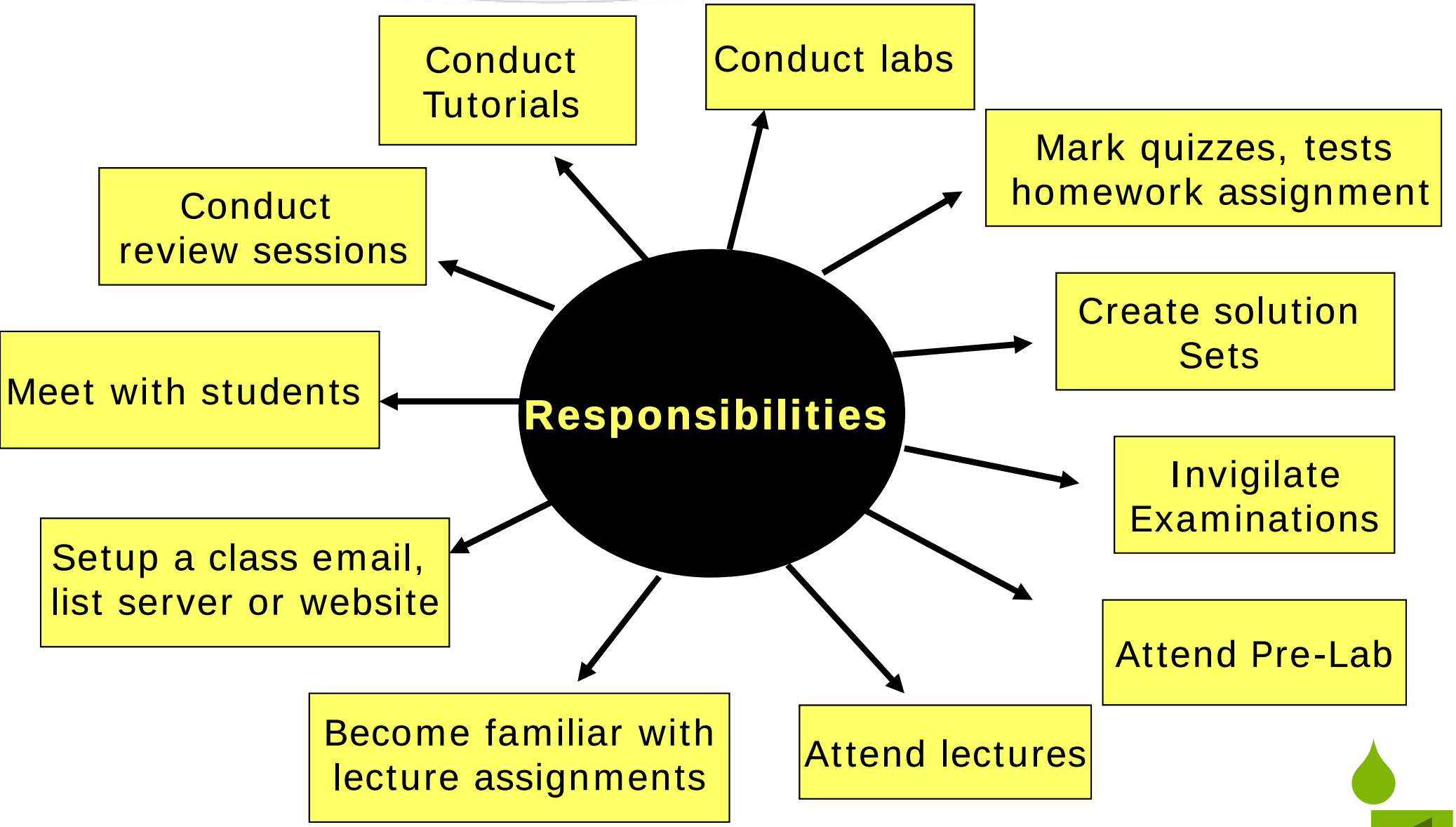
"Teaching Assistant"

- **Assist who?**
- **What're your duties?**
- **Who can assist you?**

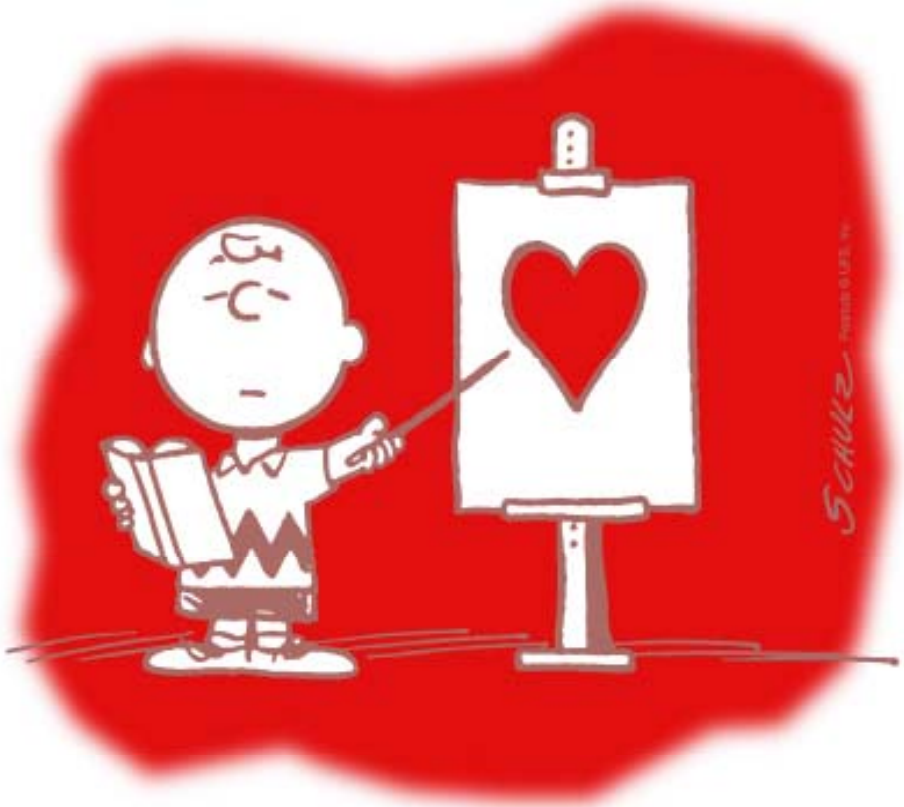
HKUST said…

- 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**. On one hand, they are still students. They shoulder the research project and take some courses themselves. On the other hand, they are trained to help instructor to **answer students' questions after the course.** During this period, they should be responsible, patient, and serious…

What are the possible responsibilities of a TA?



EFFECTIVE TEACHING SKILLS



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.**

<u>From your past experience:</u>

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

~ 5 Minutes ~

Common problems of TA

1. Lack of preparation
2. TOO MUCH
3. *Too Fast*
4. Too Boring

Lack of preparation

http://www.queercents.com/wordpress/wp-content/uploads/2007/10/climb-stack-of-paper.jpg

1. Preparation BEFORE you start

1) Clarify your responsibilities at the beginning

- Course instructor
- TAC

- The other TAs (current/previous)

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

Preparation! Preparation! Preparation!

- **1.** Do the pre-lab properly AND attend related lectures

- 

2. Read supplementary readings

- **3.** Be familiar with the Course content, administration and policy (read the course outline)

http://www.ahajokes.com/cartoon/readdum.jpg

Time Management Workshop

TA 124

Balancing Time between

TA Duties and Research

By Dr Lucia Yeung

Oct 2009

Common problems of TA

1. Lack of preparation
2. TOO MUCH
3. *Too Fast*
4. Too Boring

PHP Skinny Sheet

VisiBone

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

```
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) {}
}
```

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

Function Tools

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

Operator Precedence

```
    $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 @

     $n++;  ++$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;

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

    $b = !$b;  // boolean not

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

$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?

all these are CSCs

Time & Date

```
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;
```

Array Simple

```
$a = array(2, 'b', 'or' => 'not', array('2b'), );  // ◢ trailing comma ok
assert(4 == count($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

```
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

```
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) == $a);
assert(usort($a, create_function('$L,$R', 'return strcasecmp($L,$R);'))
    && array('a','B','c') == $a);
```

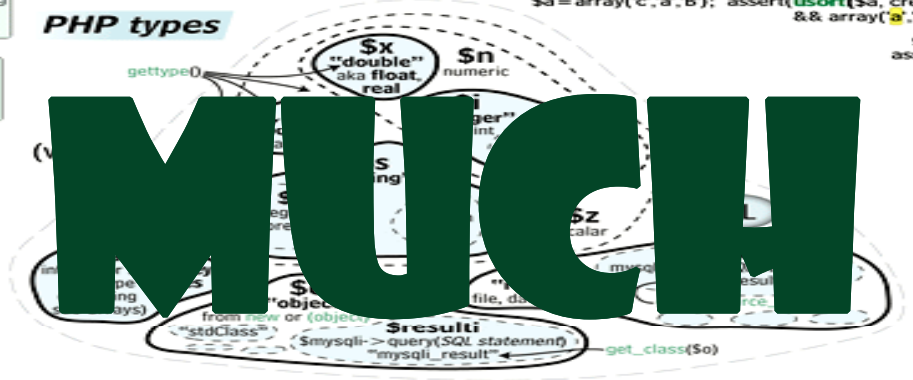
Test Test

```
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, $_POST, $_REQUEST, $_SERVER, $_SESSION;

PHP types



Forms

```
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

```
odd-looking output command with parentheses without parentheses
?>odd-looking output command<?php
echo("with ","parentheses");
echo "without ", "parentheses";
almost identical to echo()
print "almost identical to echo()";
3.14
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)));
```

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

(outputs PHP syntax)

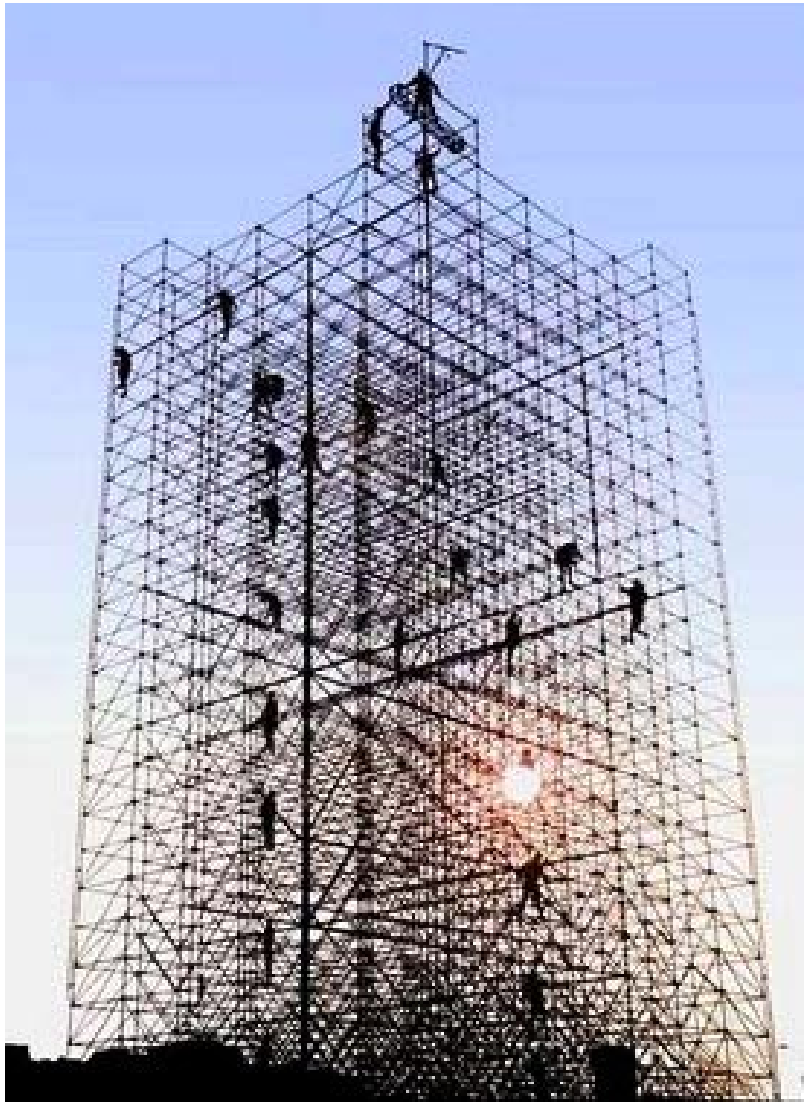
```
define('TWO', 2);  assert(defined('TWO'));  // remember quotes
hopethisreturnstrue() or die('dashed hope');
$z = array();  assert(empty($z));  $z = '';  assert(empty($z));
    guess('linux' == $_ENV['TERM']);
if (errorlevel99()) { exit(99); } else { exit('message'); }
    $a = getallheaders();  guess(preg_match(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

$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));  // ✉
ob_start();  echo 'output';  $s = ob_get_clean();  assert('output' == $s);
...a very very long report...  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('/path/file.php' == $_SERVER['PHP_SELF']);
guess('217.5.153.85' == $_SERVER['REMOTE_ADDR']);
guess('/home/username/htdocs/path/file.php' == $_SERVER['SCRIPT_FILENAME']);
$q = 2;  unset($q);  assert(!isset($q));
```

Toolbox

```
assert(4 == 2 + 2);
assert('4 == 2 + 2');  // more informative failure
define('THREE', 3);  assert(3 == THREE);
```

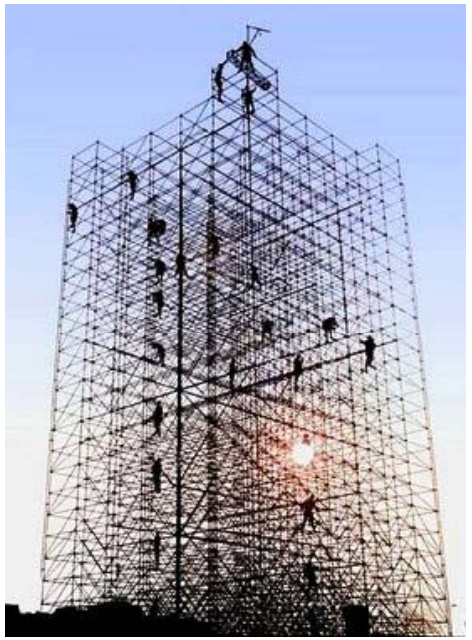
Scaffolding



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

Scaffolding

- Assess the current level of your students (e.g. by asking questions)
- Decide the next level of appropriate instruction
- Assess if students



Scaffolding

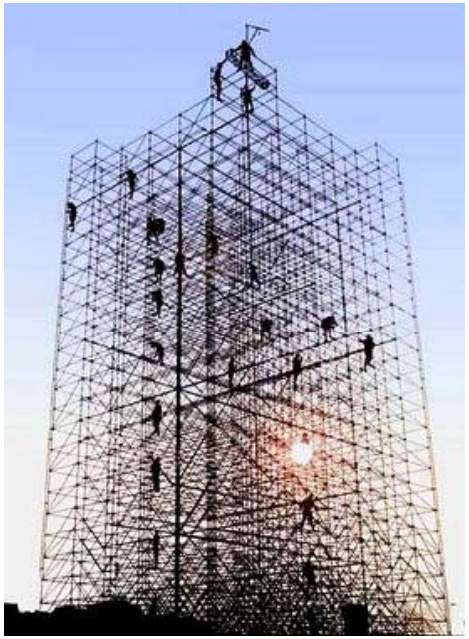
EXAMPLE:

- **Current Level:** 2 digit addition

  25+81=?

- **Next Level:** ??

- **How to assess?**



Common problems of TA

1. Lack of preparation
2. TOO MUCH
3. *Too Fast*
4. Too Boring

Too Fast

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

Too Fast

Presentation Skills

Too much information

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

Presentation Skills

- Speaker slower
- Use powerpoint / Notes
- Ask students if they have any questions

- TA113 Effective Presentation Skills
  - This is a 1.5-hour workshop mandatory 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 (Oct 12 – Mon 4:30pm)

Common problems of TA

1. Lack of preparation
2. TOO MUCH
3. *Too Fast*
4. **Too Boring**

Too Boring

http://images-0.redbubble.net/img/art/size:large/view:main/251367-7-baby-yawning.jpg

Question

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

-
-
-
-

Too Boring...

Teaching aspect

- Too much
- Too fast e.g presentation skills – FLAT tone
- Lack of Activities in class

Psychological aspect

- Problem with Values
- Lack of Attention

Too Boring...

Lack of Attention

Too Much

Lack of Motivation

Problem with Values

Presentation Skills

Lack of Interactive Activities

Interactive Teaching

- **TA 123** Techniques for Interactive Teaching (Oct 7 – Wed 4:30pm)
- This is a 1.5-hour elective workshop with an aim to enhance the effectiveness of teaching by increasing the interaction between teachers and students. We will (1) introduce the principles of interactive teaching method, (2) compare the interactive method with the traditional one, and (3) give some examples of the interactive teaching techniques.

Attention Span

- How long can you pay attention in a lecture?

Lack of Motivation

What are the reasons when you are not motivated in a class?

-
-
-
-

Potential Problem

- 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.

Examples

- **Intrinsic Value:** e.g. Newton's laws of motion; its application in mechanics.
- **Attainment Value:** e.g. Feel good
- **Utility Value:** e.g. Pass the course to graduate

Summary

1. Lack of preparation
2. TOO MUCH
3. *Too Fast*
4. Too Boring

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 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.

Available resources (cont')

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.

Any Questions?

http://202.118.40.5/y2/img/200851815224questions.gif

My Teaching Philosophy

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

Luke 6:31

The End

Thank You