Chapter 9
Module 7

MCQ Test
✔ Closed book
✔ Closed computer
✔ 20 questions
✔ Leave if finished, but come back at 1:00 for lecture

MCQ Test
✘ Any questions?

Java
✔ No more lecture material, but
✔ I will respond to questions with material as needed
✔ Following few weeks will have Java/Swing centered questions, and
✔ Assignment 3 (announced next week) will require Java/Swing, so get some practice in...
Common Gateway Interface

✔ CGI is a standard for helping web servers run external programs,

✔ and return dynamic web pages.

For example, a simple dynamic web page might return the current date and time, calculated by running the `date` program, and formatting the results as a web page.

```bash
#!/bin/sh

cat <<EOM
Content-type: text/html

<HTML><HEAD>
<TITLE>Output of data in HTML from CGI script</TITLE>
</HEAD><BODY>

<H1>Date:</H1>

EOM
date

cat <<EOM
</BODY></HTML>

EOM
```

When this script is placed in the directory `public_cgi` in your home directory on one of the UNIX systems, then you may refer to

http://www-cgi.comp.nus.edu.sg:8000/~yourid/mydate.cgi
**CGI script**

No requirement for CGI program to be a shell script.

Perl is very commonly used in this role.

It should not take too long to process.

```perl
#!/usr/local/bin/perl
print "Content-type: text/html\n\n";
print "<html><head><title>Print Environment</title></head><body>\n";
foreach $key (sort(keys %ENV)) {
  print "$key = $ENV{$key}<br>\n";
}
print "</body></html>";
```

**Environment variables**

- DOCUMENT_ROOT = /usr/local/apache/htdocs
- GATEWAY_INTERFACE = CGI/1.1
- HTTP_USER_AGENT = Mozilla/4.79 [en] (X11; U; Linux 2.2.16 i686)
- QUERY_STRING =  
- TZ = Singapore
CGI forms - GET

The form contents are found inside an environment variable called `QUERY_STRING`, as a series of name/value pairs. This mechanism is known as the GET mechanism, and a typical URL would look like this:

```plaintext
.../myform.cgi?name1=value1&name2=value2
```

POST

An alternative mechanism is the POST mechanism, in which the `STDIN` of the CGI program is used to process the form data.

CODE LISTING

```
<html><head>
  <title>Simple form</title>
</head><body>
<form action="env.cgi" method="GET">
  First Name:  <input type="text" name="First" size=30><p>
  Last Name:  <input type="text" name="Last" size=30><p>
  <select name="Home">
   <option>Singapore</option>
   <option>Malaysia</option>
   <option>Indonesia</option>
   <option>New Zealand</option>
   <option>The rest of the world!</option>
  </select>
  <input type="submit">
</form>
</body></html>
```
**CGI form**

When the form is submitted, the `QUERY_STRING` looks like this:

```
QUERY_STRING = First=Hugh&Last=Anderson&Home=New+Zealand
```

Within a CGI program, this series of name-value pairs may be used to return a dynamic web page based on this form data. 

**Perl** is a particularly useful language to use in this context - the `QUERY_STRING` can be split quickly into its component parts.

---

**Security**

✔ There are security issues with unrestricted CGI programs - since they run powerful programs (like perl and csh) with arbitrary parameters, they may be a source of (hacker) intrusion.

✔ It is for this reason that CGI usage is restricted here at NUS.

---

**PHP**

✔ PHP is a server-side scripting language.

✔ It looks very like standard HTML scripts, but the server automatically interprets the PHP.

✔ There are no enhancements needed for browsers.

---

**PHP**

✔ The two tags `<!--php and -->` start and end a PHP script, and identify a PHP code segment.

✔ The PHP code itself is a reasonably powerful programming language similar to Java, C and Perl, with functions, variables and so on.

PHP stands for PHP - Hypertext Preprocessor, a recursive acronym.
PHP

✔ Particularly useful to access databases.

✔ It is common to pair up PHP with MySQL, but PHP is not limited to one database type.

✔ For example if you wish to use PHP to access a Microsoft SQL server, you can install the ODBC support in the server machine, and access the server directly.

PHP code

```php
<?php
  $host = "example.com";
  $user = "user";
  $password = "password";

  $connection = mysql_connect($host, $user, $password); // Connect to the server
  if (!$connection) {
    die("Unable to connect to SQL server");
  }

  $db = "dbname";
  $result = mysql_select_db($db);
  if (!$result) {
    die("Unable to select database");
  }

  $sql = "SELECT COUNT(*) FROM guests";
  $result = mysql_query($sql);
  if (!$result) {
    die("Select Failed!");
  }

  $numguests = mysql_result($result, 0, 0);

  mysql_close($connection);

?>
```

PHP security

✔ PHP suffers less from the security issue than perl or csh CGI scripts do.

Java

```java
import java.applet.Applet;
import java.awt.*;
public class Lissajous extends Applet implements Runnable {
  Thread animate = null;
  double pi = 3.14159265359;
  int fx = 50;
  int fy = 100;
  int diffx = 0;
  int amp = 50;
  int phase = 0;
  int delay = 50;

  public void init() {
    resize(200, 200); // resize to fixed width,height
  }

  public void paint(Graphics g) {
    int X, Y, YY = 0,
    lastx = 0, lasty = 0,
    temp = 0, rev = 0;
    g.drawRect(0, 0, size().width - 1, size().height - 1); // outline
    if (fy < fx) {
      // frequency
      temp = fx;
      fx = fy;
      fy = temp;
      rev = 1;
    }
    for (int x = 0; x <= 360; x += 4) { // loop
      X = (int) (amp * Math.sin(x * 2.0 * pi / 360.0)); // x pos
      YY = (x * fy / fx) + phase;
      Y = (int) (amp * Math.sin(YY * 2.0 * pi / 360.0));
      if (x == 0) { lastx = X; lasty = Y; }
      if (rev == 1) { g.drawLine(lastx + 100, lasty + 100, X + 100, Y + 100); }
      else { g.drawLine(lasty + 100, lastx + 100, Y + 100, X + 100); }
      lastx = X;
      lasty = Y;
      if (rev == 1) {
        temp = fx;
        fx = fy;
        fy = temp;
      }
      phase = YY;
      /* Fix an error... phase shouldn't increase forever..... */
      if (phase < 0) { phase += 360; }
      if (phase >= 360) { phase -= 360; }
      g.drawString( fx + "\n\n\", 10, 20);
    }
  }
}
```

CODE LISTING
Lissajous1.java
This code may be found at
http://olddept.physics.upenn.edu/courses/gladney/minicourse/lectures/lecture2.html

or locally at
http://www.comp.nus.edu.sg/~hugh/Lissajous/Lissajous.html

Summary of topics

In this module, we introduced the following topics:

- Web-based application architectures
- CGI, PHP and Java