CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

Introduction to Digital Humanities Research & Computing

Fall Semester 2015

Week 9

Weekly Discussion

Perseus Digital Library

Alternative methods for writing programs

Object-Oriented Programming - a quick recap

- makes programs easier to write
- easier to understand
- easier to modify
- these advantages allow a programmer to focus more on solving problems

'Curly Bracket' Languages

- a family of related languages commonly known as 'curly bracket' languages
- curly brackets used to define start and end of a block of commands

```
#include <stdio.h>

void main()
{
   printf("Notice how the curly brackets\n");
   printf("identify the beginning and end\n");
   printf("of your commands?\n");
}
```

'Curly Bracket' Languages - C language

the power of C

- combination of assembly language options and high-level ease
- C lets you focus on the logic of a program
- often used for writing large, complicated programs such as
    - Operating Systems
    - Word Processors...
- C programs can crash other applications and the OS itself

'Curly Bracket' Languages - C language

the efficiency of C

- compiler tends to create smaller, faster, more efficient programs
- keywords are special commands used in every programming language
- the more keywords, the fewer commands you need
- more keywords can lead to a less efficient compiler & more work
- C uses libraries of sub-programs to mimic keywords in other languages

'Curly Bracket' Languages - C language

the portability of C

- C makes it easier to create compilers compared with comparative languages
- easier to compile and run on multiple computers and OSs
- portable language and programs

'Curly Bracket' Languages - C++

Adding Object-Oriented to C with C++

- object-oriented principles and benefits added to C
- more programs now being written in C++
- many learn C, and then migrate to C++ for OO principles

'Curly Bracket' Languages - Java

a few benefits and portability

- C and C++ not truly portable (minor and often major changes required)
- Java created by Sun Microsystems
    - fun timeline for Java
- Java also based on C
- Java isolates programmer from accessing computer's memory
- reduces power of Java but does translate in safer programs
- Java compiled into 'bytecode' or 'pseudocode' (p-code)
- Java Virtual Machine (Java VM)

Scripting Languages

- languages such as C and C++ often called 'system programming languages'
- scripting languages customise existing programs & work with one or more existing program
- scripting languages can work in suites such as MS Office...
- scripting languages differ from more traditional languages
    - interpreted & require source code and associated programs to run
    - typeless

Variables in PHP

- used to hold values or expressions

A few simple rules:

- variables start with $ sign
- must begin with a letter or _ character
- can only contain alphanumeric or underscore characters
- avoid spaces in the names
- names are case-sensitive

```
$course2 = “DIGH 401”;
```

- PHP is a loosely typed language

Variables in PHP

- 4 scopes for variables in PHP
    - local
    - global
    - static
    - parameter

Local

```
<?php
$a = 5; // global scope
function myTest()
{
echo $a; // local scope
}
myTest();
?>
```

Global

```
<?php
$a = 5;
$b = 10;
function myTest()
{
global $a, $b;
$b = $a + $b;
}
myTest();
echo $b;
?>
```

```
static $rememberMe;
```

```
function myTest($para1,$para2)
{
// function code
}
```

Scripting Languages

typically used in four different ways

- automate repetitive tasks
- customise the behaviour of one or more programs
- transfer data between two or more programs
- create standalone programs

Scripting Languages

1. automate repetitive tasks

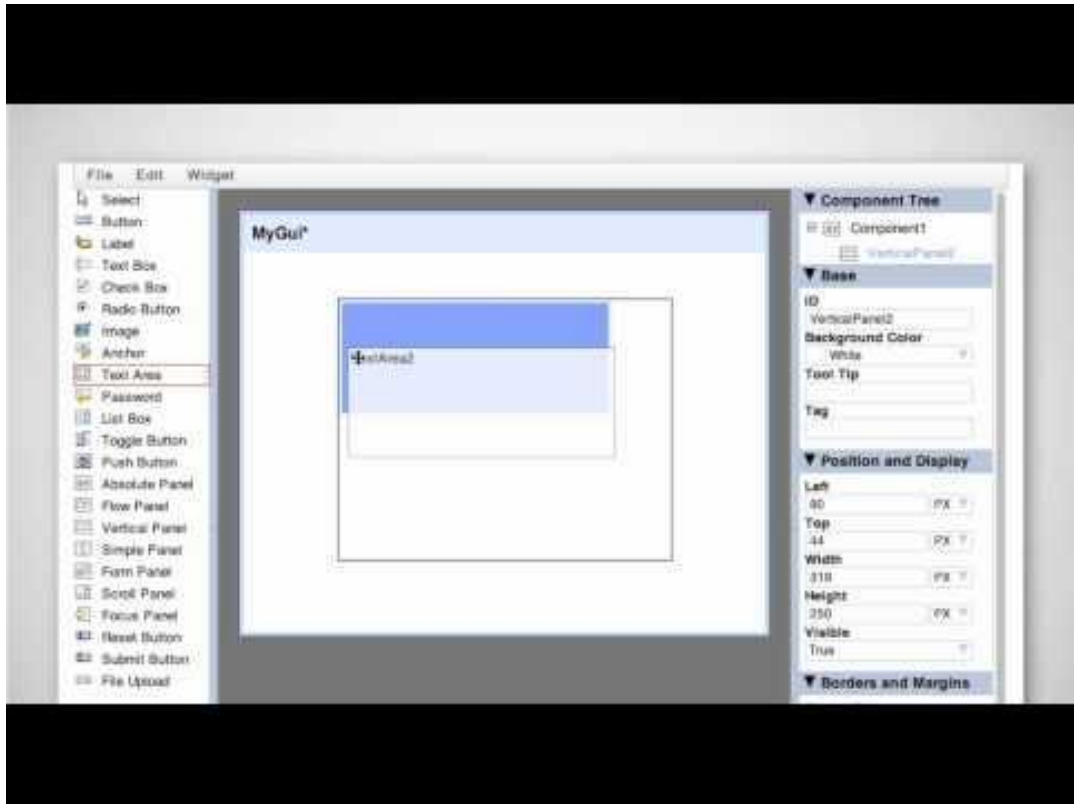
- macro to record a given task
- use macro to repeat a task

Scripting Languages

2. customise the behaviour of one or more programs

- easy to customise and reduce potential errors
- can automatically add data correctly
- combine automation and customisation eg: AppleScript

Google Apps Script


File
Edit
Widget
Select
Button
Label
Text Box
Check Box
Radio Button
Image
Anchor
Text Area
Password
List Box
Toggle Button
Push Button
Absolute Panel
Flow Panel
Vertical Panel
Simple Panel
Form Panel
Scroll Panel
Focus Panel
Reset Button
Submit Button
File Upload
MyGui*
Component Tree
Component1
Base
ID
Background Color
White
Tool Tip
Tag
Position and Display
Left
Top
Width
Height
Visible
True
Borders and Margins

Scripting Languages

3. transfer data between two or more programs

- independent scripting language such as PHP, Perl, Python, Ruby or Javascript
- these linking scripting languages are often referred to as 'glue'
- use 'glue' to combine existing programs to create custom solutions

Scripting Languages

create standalone programs

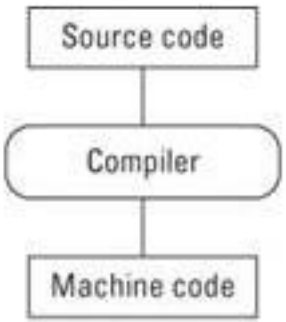
- a good example is Visual Basic
- LiveCode, or Revolution, is another popular example
- interpreter for LiveCode allows programs to run on different OSs

Programming Tools

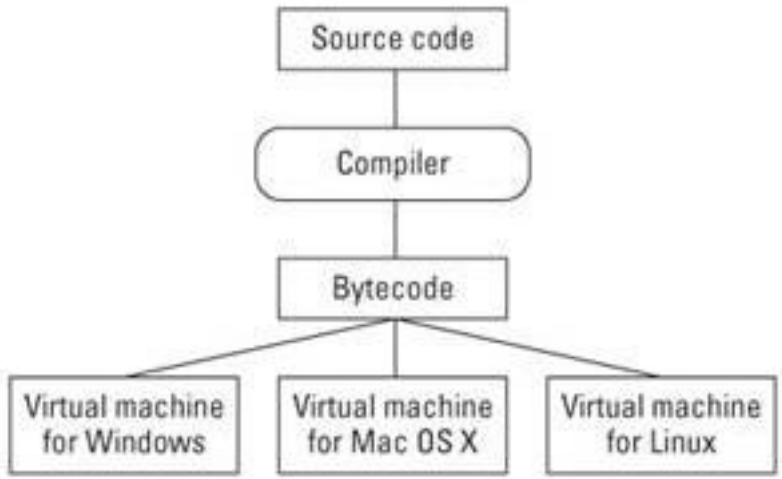
- an editor and compiler or interpreter
- compiler converts source code into machine code
- machine code saved as an executable
- debugger and profiler
- disassembler

Programming Tools - Virtual Machine

- compilers are often difficult to make for multiple OSs and processors
- interpreters need the source code to run
    - often unsuitable for broadly distributing software
- virtual machine was designed to address these issues
- bytecode or pseudocode (p-code)



A compiler normally converts source code directly into machine code for a specific type of processor.



When compiled to bytecode, a program can run on any operating system that has the bytecode virtual machine installed.

Programming Tools - Virtual Machine

- reverse engineer from bytecode to source code
- Java currently most popular language to use a VM
- Java compiles source code into bytecode and runs using a VM

Programming Tools - Sandbox

- testing environment that isolates untested code changes, experiments...
- protects live and active servers and their data
- replicate at least minimal functionality to enable testing and development...
- version control software such as Mercurial, Subversion, and Git

Bitbucket | GitHub

The act of digitisation

Digitising Text

- image of a printed page
    - scan or photograph a page of text

- advantages might include:
    - relatively quick and ease to produce
    - close representation of original
    - add tools for ease of manipulation

- disadvantages might include:
    - limited search options within image
    - file size is considerably larger compared to text
    - hardware considerations

The act of digitisation

Digitising Text - shall we use markup?

- another option for digitisation of textual material
- advantages such as complete machine readability
- markup may take many different forms
- all deal with the classification of components of a document
- more sophisticated capture structural and descriptive aspects

"explicit (to a machine) what is implicit (to a person)"
(Burnard, Lou. "Digital Texts with XML and the TEI". Text Encoding Initiative)

The act of digitisation

Digitising Text - conforming to a standard

- exciting opportunities are possible when we all conform to a standard
- document markup predates the internet and computers
- separation of content from format
- GML and SGML
- open and flexible but also complicated, time-consuming and expensive
- conforming to the concept of TEI

The act of digitisation

Digitising Text - Options for making your own digital text

- import, automate, or type your own
- scan or photograph analogue to digital
- minimum 300dpi scan or better 600dpi
- book scanners and cradles are also available
- OCR texts
- OCR limitations include poor performance for
    - non-Latin characters
    - small print
    - certain fonts
    - complex page layouts or tables
    - mathematical or chemical symbols
    - most texts pre-19th century

The act of digitisation

Digital Images

- again we can use scanners and cameras
- more specialised material with cradle and book scanners
- consider fidelity to the original and long term preservation
- quality of digital result dependent upon
    - quality of original
    - digitising method employed
    - sample rate of digital to analogue
- minimum 300 DPI, preferably 600 DPI
- uncompressed format, such as TIFF
- optimise to JPEG for online publication

The act of digitisation

Digital Images - metadata

- metadata added to images
- in particular important when the files are not text searchable
- standards such as Dublin Core or METS
- technical information about the file such as
    - resolution
    - compression (if applicable)
    - scanning process
    - copyright and access rights
    - authorship of the image
    - and much more...

Weekly Exercise - Example Solution 1

```
<?php
$march = 31;
$april = 30;
$july = 31;
$august = 31;
$december = 31;
$subtotal;

function addDays() {
global $march, $april, $july, $august, $subtotal;
$subtotal = $march + $april + $july + $august;
}

function minusDays($days) {
global $subtotal, $december;
$subtotal = $days - $december;
}

//run function to calculate adding days
addDays();

//output first subtotal to check addition function
echo '<p>first subtotal = '.$subtotal.'</p>';

//run function to calculate subtracting days
minusDays($subtotal);

//output result of calculations
echo '<p>final subtotal = '.$subtotal.'</p>';
?>
```

- how can we improve this solution?
- any limitations?
- abstraction issues?

Demo

Weekly Exercise - Example Solution 2

```
<?php
$subtotal;

function addDays() {
global $subtotal;
$march = 31;
$april = 30;
$july = 31;
$august = 31;
$subtotal = $march + $april + $july + $august;
}

function minusDays($days) {
global $subtotal;
$december = 31;
$subtotal = $days - $december;
}

/**run function to calculate adding days**/
addDays();

/**output first subtotal to check addition function**/
echo '<p>first subtotal = '.$subtotal.'</p>';

/**run function to calculate subtracting days**/
minusDays($subtotal);

/**output result of calculations**/
echo '<p>final subtotal = '.$subtotal.'</p>';
?>
```

- what's wrong with this solution?
- what are its limitations?
- abstraction?

Demo