CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

DIGH 401 - Introduction to Computing

Fall Semester 2014

Week 4

Today's Class

Weekly Exercise

Programming Languages

- assessment of 'Curly Bracket' languages
- scripting languages
- variables etc

Programming tools

Weekly Exercise

Please choose a favourite graphical user interface for an electronic gadget, for example a smartphone, computer operating system, video game etc. This device does not have to be current, it could be a classic/vintage gadget, but it does require a demonstrable graphical user interface.

Consider this interface for a moment from a user's perspective, and then outline a conceptual code design for the development of this user interface. You may use either structured or object oriented programming design, but be prepared to justify your choice of pattern.

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

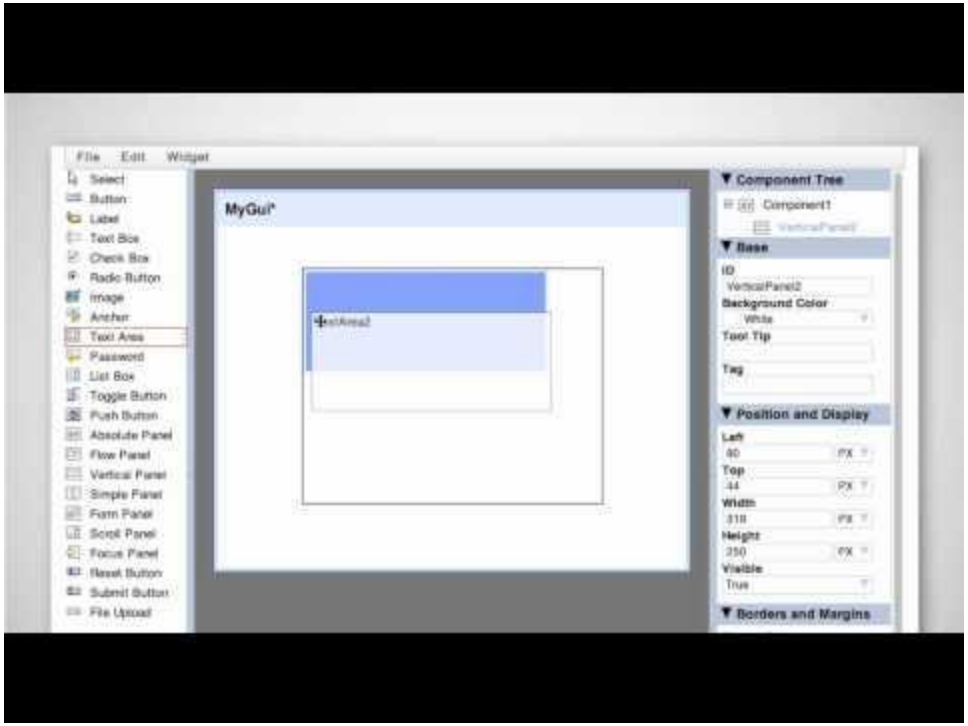
Scripting Languages

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

DIGH 401 - Introduction to Computing



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

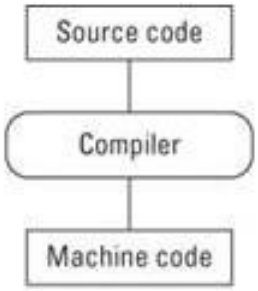
- a good example is Visual Basic
- LiveCode, or Revolution, is another popular example
- interpreter for LiveCode allows programs to run on different OSs

Programming Tools

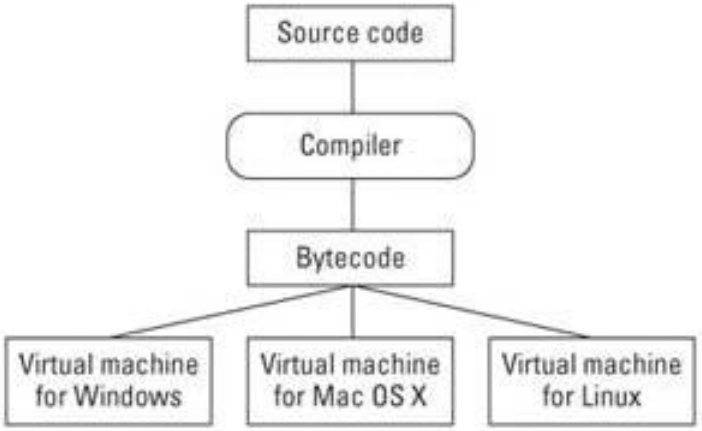
- an editor and compiler or interpreter
- compiler converts source code into machine code
- machine code saved as an executable
- debugger and profiler
- disassembler

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

Google Code | GitHub