CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

Introduction to Digital Humanities Research & Computing

Fall Semester 2015

Week 6

Discussion

Newton Project

Getting started with programming

- what do you need to get started?

    - choose an OS
    - choose a preferred editor
    - select your programming language
    - compiler will be relative to the selected language

Getting started with programming

eg: Java programming

- A Java compiler and virtual machine
    - available from www.java.com
- An editor such as Eclipse
    - available from www.eclipse.org

Getting started with programming

eg: PHP programming

- a basic text editor to create and save .php files
    - eg: TextWrangler or BBEdit on OS X...

- a PHP enabled web server, such as Apache 2 with PHP support
    - XAMPP

- a web browser to view the output

Getting started with programming

- describe how programming works

- employ examples from various different languages

- begin practicing and initial programming with a specific language

Getting started with PHP programming

- PHP:Hypertext Preprocessor

- server-side language
  - PHP code is executed on the server

- support for many different databases including MySQL, PostgreSQL...

- open source software
  - free to download and use
  - included with LAMP stacks, XAMPP test packages...
  - http://www.php.net

Getting started with PHP programming

- runs on multiple platforms including Linux, OS X, Windows, Unix...

- compatible with many web servers such as Apache, IIS...

- files can include different content such as
    - text, HTML tags, scripts, styling...

- PHP files are interpreted by a server and returned to the browser as plain HTML

- files normally end in .php

Getting started with PHP programming

```
<?php
?>
```

- can be stand-alone or embedded in HTML, for example

```
<html>
<body>
<?php
echo "Hello World, how are you?";
?>
</body>
</html>
```

- each code line or grouping must end with a semi-colon
    - distinguishes one set of instructions from another

Alternative methods for writing programs

Spaghetti programming - no real plan

- code first, ask questions later
- modification increases the requirement for ease of understanding
- take care how and where you modify code
- no structure = mess of code
- large number of collaborators compounds this problem

Alternative methods for writing programs

Spaghetti programming - GOTO command

- BASIC programming language most closely associated with Spaghetti programming
- GOTO commands told the computer to 'go to' another part of the program
- the code jumps from one part to another

Alternative methods for writing programs

Spaghetti programming - GOTO command

```
10 GOTO 50
20 PRINT "THIS LINE PRINTS SECOND"
30 END
40 GOTO 20
50 PRINT "THIS LINE PRINTS FIRST"
60 GOTO 40
```

- the code becomes increasingly hard to read, modify, and understand

Alternative methods for writing programs

Structured programming - planning ahead

- keep a program organised from the start

- teaches programmers that a program can be divided into 3 distinct parts

- SEQUENCES
- BRANCHES
- LOOPS

Alternative methods for writing programs

A PHP loop example:

```
for ($x=1; $x<=5; $x++)
{
    echo ‘X = ’.$x.’<br>’;
}
```

DVD playback options for a movie using a

- sequence
- branch
- loop

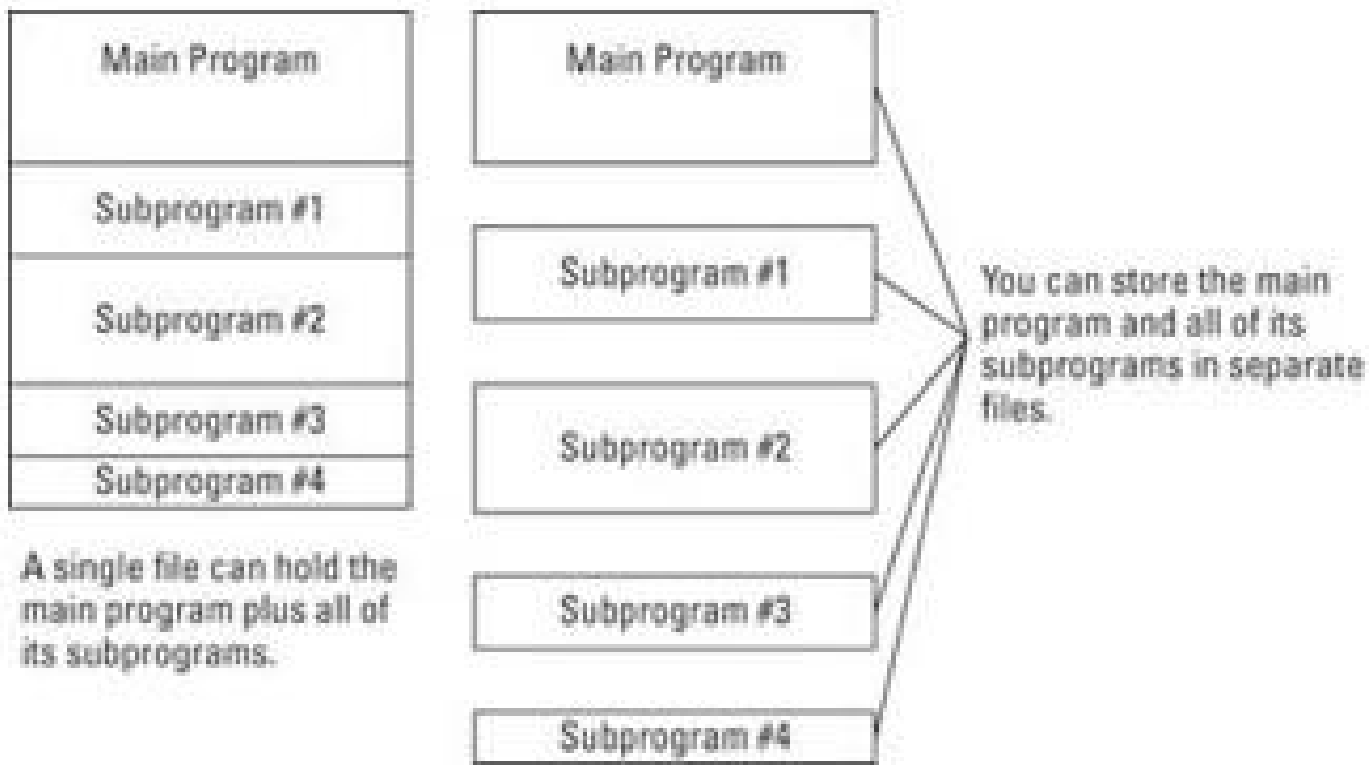
Alternative methods for writing programs

Top-down programming

- dividing a large program into smaller parts
    - easier to manage
    - each part performs a specific task
- identify main (top) task for the program to solve
- identify smaller sub-programs within the larger program
- smaller modules building the larger whole
- larger program consisting of many subprograms
- store subprograms in either
    - one file
    - separate, multiple files (this is often the preferred option)

Alternative methods for writing programs

Top-down programming



Alternative methods for writing programs

Object-Oriented Programming

- Java is an Object-Oriented Programming (OOP) language
- another technique for dividing large programs into manageable parts
- solves two obvious issues with structured programming
    - reusability
    - modelling
- reusability in OOP with objects

Alternative methods for writing programs

The issue of landing a spaceship on the moon!

NASA orbiter error

Alternative methods for writing programs

Object-Oriented Programming - Objects and division

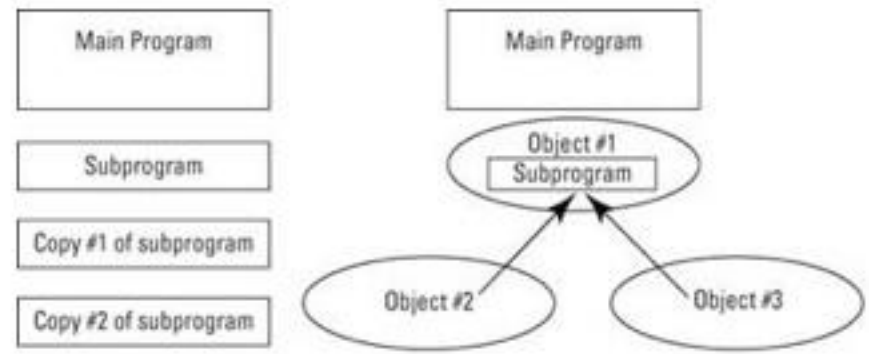
- OOP avoids this issue by issuing objects
- objects combine data and the commands that manipulate them
- OOP divides a large program into real life objects

Think about landing on the moon using an OOP design.

Alternative methods for writing programs

Object-Oriented Programming - Objects and modifications

- objects simplify modification
- objects also permit code reusability
- more efficient and easier than sharing sub-programs
- sub-programs can compound errors
- inheritance in OOP
- OOP never physically copies a sub-program but 'points to' or 'inherits'



Copying a subprogram creates multiple copies of that subprogram.

Instead of making copies of a subprogram, objects "inherit" a subprogram. This leaves a single copy of a subprogram that can be used in multiple objects.

Alternative methods for writing programs

Object-Oriented Programming

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

Online research exercise

Jules Verne and H.G.Wells

- find as many different editions of

- Around the World in Eighty Days
- Twenty Thousand Leagues Under the Sea

- The Time Machine
- The War of the Worlds

- record availability, copyright notice, PDF download option, original source for digitised copy...

- create a collaborative spreadsheet and share with group

Scifi Authors website