CENTER FOR TEXTUAL STUDIES AND
DIGITAL HUMANITIES

DIGH 401 - Introduction to Computing

Fall Semester 2014

Week 3

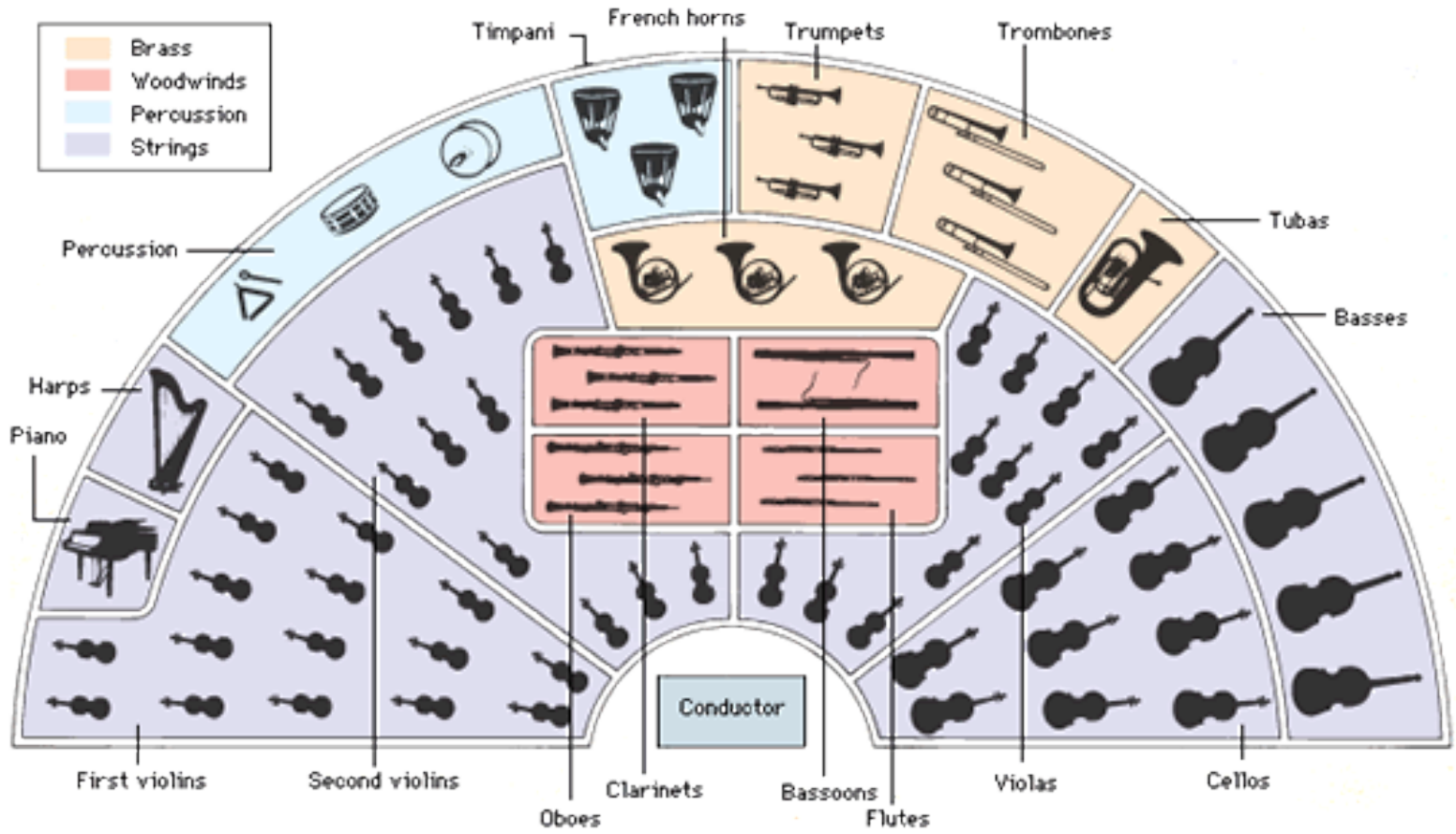
Today's Class

- Last week’s exercise

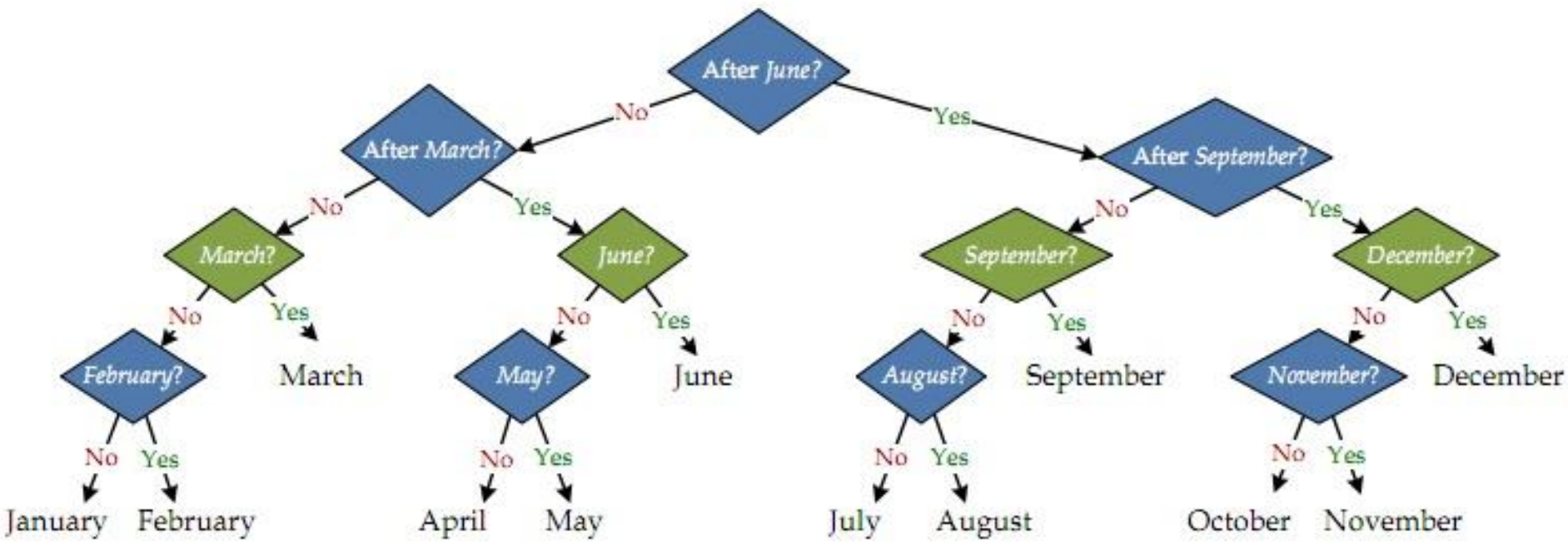
- Alternative methods for writing programs
    - with no discernible plan (spaghetti programming)
    - planning ahead (structured programming)
    - organising a program (Object Oriented programming)

Class Exercise

conceptual algorithm and binary tree to calculate the chosen instrument of a musician in an orchestra



Sections of an orchestra
(shamelessly pilfered from a search of Google Images)


After June?
No
Yes
After March?
After September?
No
Yes
No
Yes
March?
June?
September?
December?
No
Yes
No
Yes
No
Yes
No
Yes
February?
March
May?
June
August?
September
November?
December
No
Yes
No
Yes
No
Yes
No
Yes
January
February
April
May
July
August
October
November

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

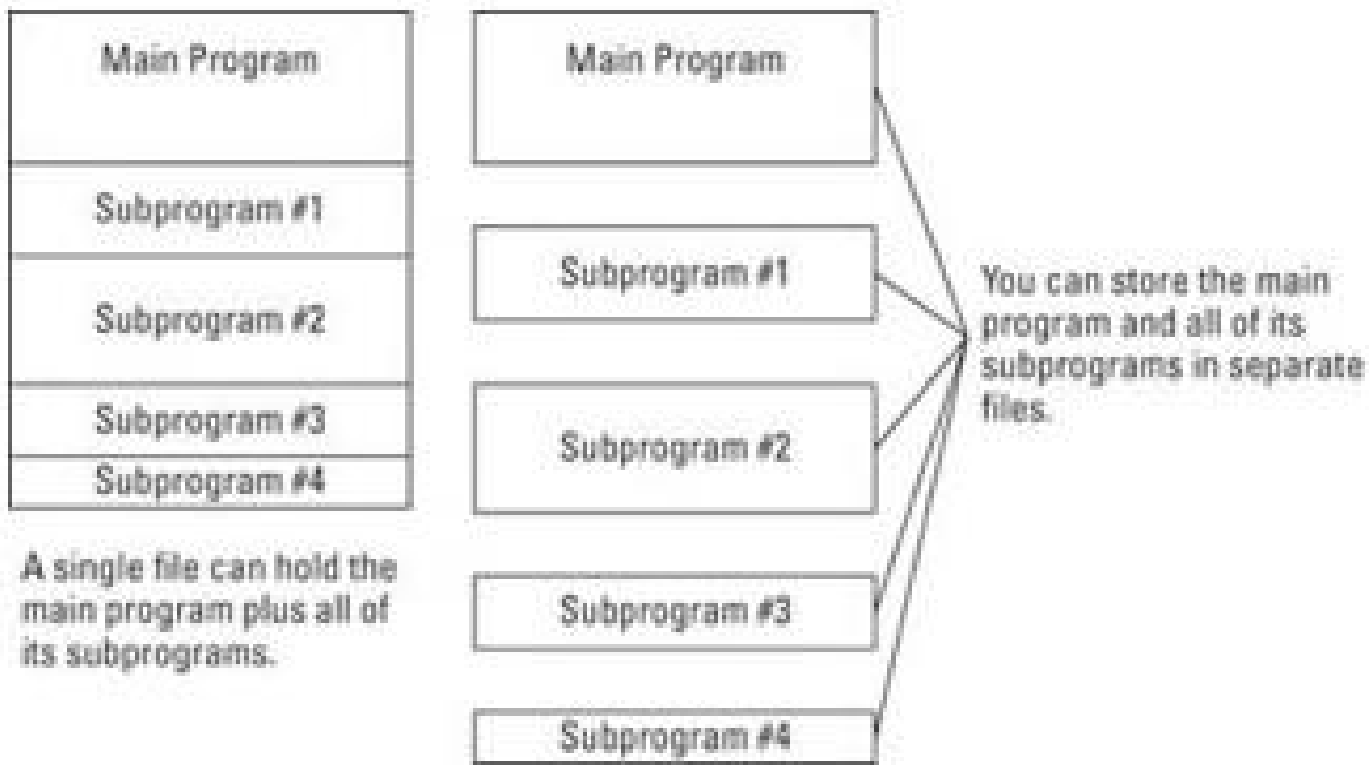
Alternative methods for writing programs

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

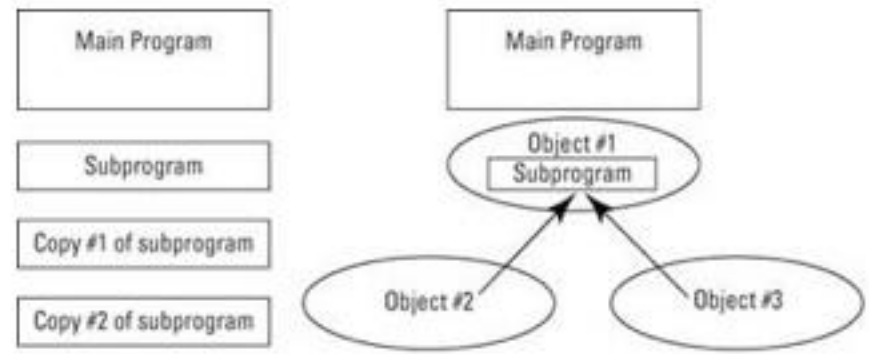
- OOP avoids this issue by issuing objects
- objects combine data and the commands that manipulate them
- OOP divides a large program into real life objects

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