

Group Assessment & Design Patterns

IMA Workshop June 2019

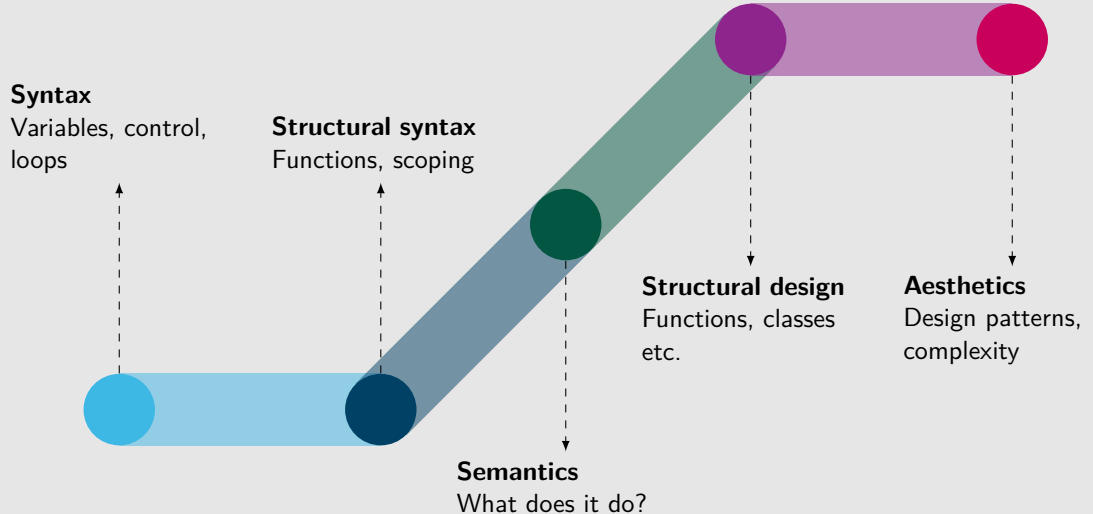
Matthew M. Jones

- Programming as a problem solving activity
- Practical software development skills using group assessment as a driver



- Second year module: **Problem Solving Methods**
- Students learn **R** (for BSc Maths students), and **Java** (for BSc Maths with Computing - having learned **Python** in year 1)
- They arrive in the second year with good knowledge of syntax and coding

The software development journey



- Workshops run entirely using a problem based approach
- Programming challenges build on problems
- **Aim:** programming as an essential tool in mathematics

- Small scale problem repositories are common:
 - `leetcode.com` - interview problems from big software companies
 - `codewars.com`
 - `topcoder.com`
 - `coderbyte.com`

Most in demand hard skills

1. **Cloud Computing**
2. **Artificial Intelligence**
3. Analytical Reasoning
4. People Management
5. **UX Design**
6. **Mobile Application Development**
7. Video Production
8. Sales Leadership
9. Translation
10. Audio Production
11. **Natural Language Processing**
12. **Scientific Computing**
13. **Game Development**
14. Social Media Marketing
15. Animation
16. **Business Analysis**
17. Journalism
18. Digital Marketing
19. Industrial Design
20. Competitive Strategies
21. **Customer Service Systems**
22. **Software Testing**
23. **Data Science**
24. **Computer Graphics**
25. Corporate Communications

Source: The Most In-Demand Hard and Soft Skills of 2019, Gregory Lewis
LinkedIn Talent Blog, January 3, 2019

Application Programming Interfaces

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

[Run code now](#)[Try in Google's interactive notebook](#)

- software development is increasingly about accessing APIs
- so what is the future for mathematicians who can programme?




Team work in programming

- Team work provides huge potential for learning software development
- Team work encourages students to solve the big problems in collaboration:
 - Version control
 - Compatibility
 - Working and writing APIs




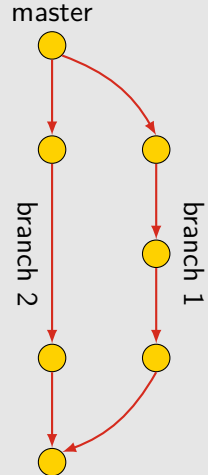
- Object Oriented but also aspects of Procedural and Functional paradigms
- A very well established language, very employable, very fast, but very verbose
- It is a **BIG** language

Teaching the practicalities

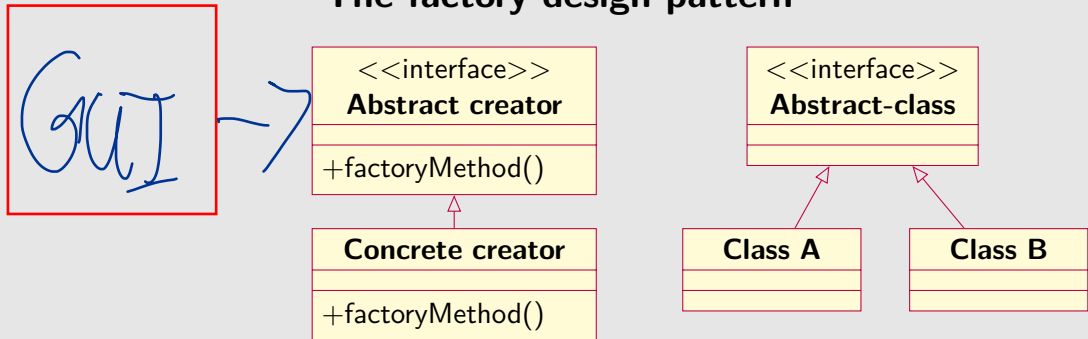
- APIs, REST(ful) APIs, Web Applications, etc etc
- Version control:  **git** , and  GitHub
- Build and dependency management tools: **maven** , or  Gradle

Working in teams: an example

-  git
- Central repository (on GitHub) – students **pull** the software to a local repository
- When finished they can **commit** and **push** their code up to the central repository
- Git uses a branching model to ensure changes don't damage the **master** version



The factory design pattern



Working in teams: an example

- API is separated from client
- The design means that each of the classes can be coded by entirely different people

- Being a confident programmer means more than learning to code
- A problem based approach provides an opportunity to stimulate students' interest in software development
- Group assessment is a fantastic opportunity to introduce students to more sophisticated ideas