# Chapter 1

## Goals

Write correct and maintainable software:

### Correctness

It follows the specifications.

#### How do we achieve correctness?

1. Principled design and development
2. Abstraction and modularity
3. Documentation

#### How do we verify correctness?

1. Reasoning about code
2. Formal software verification
3. Testing

### Maintainability

1. Using object oriented models
2. Polymorphism – When a class whose operations can be applied to different types?
3. Principled design and development
4. Abstraction and modularity
5. Documentation

## Software Engineering

About mitigating and managing complexity

Managing change

Dealing with software failures

## Version Control

### Steps:

1. Checkout / clone set of files to a local repository
2. Add files to version control
3. Commit / Push files to add changes to the master repository
4. Update / Pull changes from master repository

### Merges

Occur when multiple users have modified the same file – creates a conflict

To avoid, always Pull before pushing.

## Testing