Lecture_14

Software Implementation

Switch Statements

- Use of complex switch -case and if then statements
  - Multiple conditions
  - Likely more conditions will be added
- Common refactoring techniques
  - Extract method, replace type code with subclasses, replace conditional with polymorphism
  - **Polymorphism** - Make class more abstract, add sub classes

```C++
void calcPrice(cart){
        for(int i =0; i < cart.length){
                if(cart[i].catagory == "food")
                // etc
        }
// Rather than doing this we should move these to a function

for(int i =0; i < cart.length){
                total += calcPrice;
                // etc
}
```

Tools to Identify code smells

- Maintainability rating
- Between 0% to 5%
- Between 6% to 10%, B
- Between 11% to 20%, C
- Between 21% to 50%, D
- anything over 50% is an E

Code Styles

- Common format that has been shown to reduce the time it takes a developer to understand a piece of code
- Not affect functionality, just maintainability, readability, and consistency
- Checking in unformatted code is a form of technical debt that can lead to increases in software cost

- One statement per line
- Use indentation
- Better, use a pretty printer
- Use plenty of blank lines
  - Breaking up code

```C++
// Bad example
for(auto j: auto k){if(j == 1){return false}}
```

Remarks on Programming Standards

- The aim of standards is to make maintenance easier
  - If they make development difficult, then they must be modified
  - Overly restrictive standards are counter productive