Lecture 1

New topics include the `explicit` keyword, the `IntCell` class, initialization with brackets {}, range loops, lvalues, and rvalues. As for style, functions and classes follow CamelCase, variables follow snake_case

IntCell class

Just holds an int (accessor and setter)

Proper ways to construct:

- `IntCell obj1;`
- `IntCell obj2(12);`
- `IntCell obj3{12};`
- `IntCell obj4{};`

Vector Initialization

- `vector<int> n = {1, 2, 4, 4};`
- `vector<int> n {1, 2, 3, 4};`
- `vector<int> n(12); -- empty vector of size 12`
- `vector<int> n{12}; -- vector containing element 12`

Range based For loops

```
vector<float> f = {1.1, 10.2, 3, 20.31};

float sum = 0;

for (float x: f)

    sum += x;
```

- x cannot be modified (it's a copy)

lvalues and rvalues

- lvalues are containers, rvalues are things that are contained
- lvalues point to a specific place in memory, rvalues may be stored in a register temporarily but otherwise are not stored
- lvalues are typically the declaration / variable on the lefthand side, while rvaues are typically expressions / literals on the righthand side

examples

lvalues

- `const int x;`
- `int y;`
- `vector<string> arr(3);`

rvalues

- `2`
- `x + y`
- `'foo'`