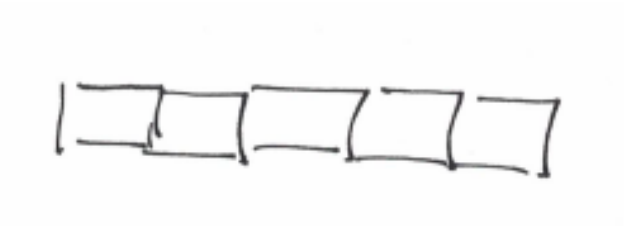
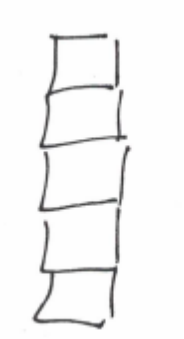
CSE 330 – Winter 2020 – Lecture Notes – Week 1

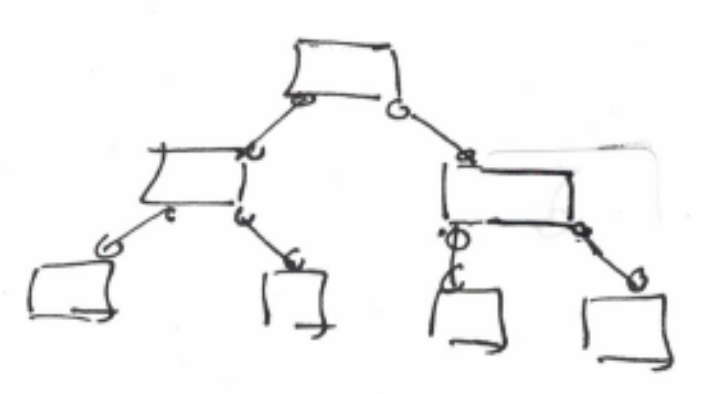
Instructor: Kerstin Voigt

Data structures are structures that contain data.

There are many ways to contain …



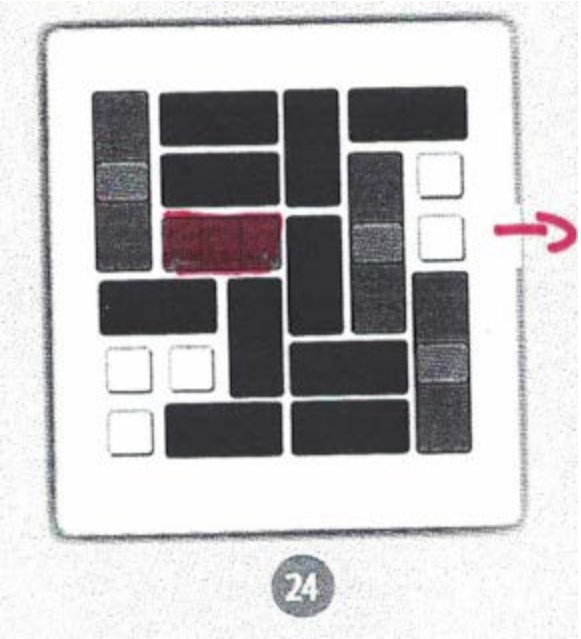




Important Questions:

- Why have different containers?
- Why contain data in the first place?
- Which container is better? ("better" = ?????)
- Which one is better for what?
- What do we want to do with the container?
- What do we want to with the data?
- What are the costs? ("cost" = ?????)

  … to the programmer
  … to the user
  … to the program



Warming up to C++…

… adopted from and/or inspired by Weiss, Chapter 1

Defining and implementing special types of integer values. Presented as a simple example (… and mostly overkill if we really only care about integer values).

```
class IntCell0
{
public:
    int storedValue;
};
```

What do we think about this? Discuss …

Making improvements:

```
class IntCell1
{
public:
    IntCell1(int initVal = 0)
        :storedValue(initVal) {}

    int read() const
    { return storedValue; }

    void write(int x)
    { storedValue = x; }

private:
    int storedValue;
};
```

More features:

```
class IntCell2
{
public:
     IntCell2(int initVal = 0)
          :storedValue(initVal) {}

     IntCell2(const IntCell2 & other)
          :storedValue(other.storedValue) {}

     IntCell2& operator =(const IntCell2& other)
     {
          storedValue = other.storedValue;
          return  *this;
     }

     bool operator ==(const IntCell2& other) const
     {
          return  storedValue == other.storedValue;
     }

     bool operator < (const IntCell2& other) const
     {
          return  storedValue < other.storedValue;
     }

     int read() const
     { return storedValue; }

     void write(int x)
     { storedValue = x; }

private:
     int storedValue;
};
```

Another way of doing things …

```
class IntCell3
{
public:
     IntCell3(int initVal = 0)
     { storedValue = new int(initVal); }

     ~IntCell3()
     { delete storedValue; }

     IntCell3(const IntCell3& other)
     { storedValue = new int(*other.storedValue); }

     /*
     IntCell3(IntCell3&& other)
          :storedValue(other.storedValue)
     { other.storedValue = nullptr; }
     */

     IntCell3 & operator =(const IntCell3& other)
     {
          if (this != &other)
               *storedValue = *other.storedValue;
          return *this;
     }

     /*
     IntCell3  operator =(IntCell3&& other)
     {
          std::swap(storedValue,other.storedValue);
          return *this;
     }
     */

     bool operator ==(const IntCell3& other) const
     {
          return  *storedValue == *other.storedValue;
     }
```

```
    bool operator < (const IntCell3& other) const
    {
        return  *storedValue < *other.storedValue;
    }

    int read() const
    { return *storedValue; }

    void write(int x)
    { *storedValue = x; }

private:
    int *storedValue;  // datamember is pointer!
};
```

Now testing:

```
int main()
{
    IntCell0 ic_0;
    ic_0.storedValue = 5;
    cout << ic_0.storedValue << endl << endl;

    IntCell1 ic_1;
    cout << ic_1.read() << endl;
    ic_1.write(12);
    cout << ic_1.read() << endl << endl;

    IntCell2 ic_2a(7);
    IntCell2 ic_2b(ic_2a);
    IntCell2 ic_2c;
    cout << ic_2a.read() << " "
         << ic_2b.read() << endl;
    cout << "Are 1st and 2nd values equal? - Answ: "
          << (ic_2a == ic_2b) << endl;
    ic_2b.write(11);
    ic_2c = ic_2b;
    cout << ic_2a.read() << " " << ic_2b.read() << " "
         << ic_2c.read() << endl;

    cout << "Is 1st less than 2nd? - Answ: "
          << (ic_2a < ic_2b) << endl;
    cout << "Is 2nd less than 3rd? - Answ: "
          << (ic_2b < ic_2c) << endl << endl;
```

```
      IntCell3 ic_3a(33);
      IntCell3 ic_3b(ic_3a);
      IntCell3 ic_3c;
      cout << ic_3a.read() << " "
           << ic_3b.read() << endl;

      cout << "Are 1st and 2nf values equal? - Answ: "
           << (ic_3a == ic_3b) << endl;
      ic_3b.write(55);
      ic_3c = ic_3b;
      cout << ic_3a.read() << " " << ic_3b.read() << " "
          << ic_3c.read() << endl;
      cout << "Is 1st less than 2nd? - Answ: "
           << (ic_3a < ic_3b) << endl;
      cout << "Is 2nd less than 3rd? - Answ: "
           << (ic_3b < ic_3c) << endl;
      /*
      IntCell3 ic_3d(IntCell3(222);
      ic_3a = IntCell3(555);
      cout << "Finally: " << ic_3d.read() << " " << ic_3a.read()
<< endl << endl;
      */

      return 0;
}
```