Strings

Victor Eijkhout and Carrie Arnold and Charlie Dey

Fall 2017

Characters

Characters and ints

- Type *char*;
- represents '7-bit ASCII': printable and (some) unprintable characters.
- Single quotes: `char c = 'a'`
- Equivalent to (short) integer: `'x'-'a'` is distance `a--x`

Exercise 1

Write a program that accepts an integer $0 \cdots 26$ and prints the so-manieth letter of the alphabet.

Extend your program so that if the input is negative, it prints the minus-so-manieth uppercase letter of the alphabet.

Strings

String declaration

```
#include <string>
using namespace std;

// .. and now you can use ‘string’
```

(Do not use the C legacy mechanisms.)

String creation

A *string* variable contains a string of characters.

```
string txt;
```

You can initialize the string variable (use `-std=c++11`), or assign it dynamically:

```
string txt{"this is text"};
string moretxt("this is also text");
txt = "and now it is another text";
```

Concatenation

Strings can be *concatenated*:

```
txt = txt1+txt2;
txt += txt3;
```

String is like vector

You can query the *size*:

```
int txtlen = txt.size();
```

or use subscripts:

```
cout << "The second character is <<" <<
        txt[1] << ">>" << endl;
```

More vector methods

Other methods for the vector class apply: `insert`, `empty`, `erase`, `push_back`, et cetera.

`http://en.cppreference.com/w/cpp/string/basic_string`

Exercise 2

Write a function to print out the digits of a number: 156 should print `one five six`. Use a vector or array of strings, containing the names of the digits.

Start by writing a program that reads a single digit and prints its name.

For the full program it is easiest to generate the digits last-to-first. Then figure out how to print them reversed.

Exercise 3

Write a function to convert an integer to a string: the input 205 should give `two hundred fifteen`, et cetera.