CS 246 Fall 2019 — Tutorial 3

September 25, 2019

Summary

1 Strings

The type for strings in C++ is `std::string`. It is included in the header `<string>`. Strings have:

- indexed access using `[]` or `at()`.

- concatenation using `+`, `+=` (both with `string` and with C-style strings). For `+`, at least one side must be a `string`. For `+=`, it must have a `string` on its left.

- lexicographical comparison using `==, !=, <, >, <=, >=`

- others: `length, clear, substr, find`

Use the `c_str()` member function to access a C-style version (`const char *`) of the string.

2 Streams

Streams are the C++ way of handling input/output.

Streams break into two classes: input streams and output streams. These streams support some common base set of functions.

2.1 Input Streams

An input stream is a stream from which information may be read. By default, reading from an input stream is whitespace-delimited.

2.2   Output Streams

An output stream is a stream to which information may be sent. Each output stream possesses a shovel operator <<; for example,

```
my_output_stream << some_data; // Put the contents of some_data into my_output_stream
```

2.3   Standard Streams

C++ has three standard streams, for input (`cin`), output (`cout`), and error (`cerr`). These are included in the `<iostream>` header.

Exercise: Write the program `printer` from the first tutorial. It should take a single integer N from standard input, and print the numbers $0, \ldots, N$, with even numbers going to standard output and odd numbers to standard error.

2.4   File Streams

Filestreams are contained in the header `<fstream>`. The two most commonly used kinds of filestreams are `ifstream`s and `ofstream`s.

Exercise: Let's write a C++ program `pseudoCat`. `pseudoCat` will take a list of filenames from standard input, separated by arbitrary whitespace. Then, it will print the contents of those files to standard output.

2.5   String Streams

String streams are included in the header `<sstream>`. String streams are streams in which formatted information can be stored, and from which the information may be retrieved.

Exercise: Create a program which accepts as input two complex numbers given as strings in the form `a+ib`, `c+id`, and produces their product `(ac-bd)+i(ad+bc)`

3   Overloading

Each function has a *signature*, which consists of the types it takes as inputs. In C++, we can have multiple functions with the same name as long as the functions have different signatures. This is called *overloading*.

Exercise: Are the following functions overloads of each other?

```
int foo() {                              std::string foo() {
    return 5;                                return "five";
}                                        }
```

4 Default Parameters

The parameters of a function can be given default values. For example,

```
void foo(int n = 75);
```

There are now two ways to call `foo`:

```
foo();    // n is 75 within foo
foo(10); // n is 10 within foo
```

In a function declaration, all default variables must come last.

Example:

```
void foo(int n = 75, char c); // invalid
void foo(int n = 75, char c = 'a'); // ok
```

Exercise: Which of the following is **not** a valid overload of `bool foo(int x, char c);`?

1. `int foo();`

2. `char foo(char x, int c);`

3. `bool foo(int c);`

4. `int foo(int x, char c, int y = 10);`

5. None of the above.