Class inheritance: is-a

Victor Eijkhout and Charlie Dey

spring 2017

General case, special case

You can have classes where an object of one class is a special case of the other class. You declare that as

```
class General {
protected: // note!
 int g;
public:
 void general_method() {};
};
class Special : public General {
public:
  void special_method() { g = ... };
};

int main() {
  Special special_object;
  special_object.general_method();
}
```

Inheritance: derived classes

Derived class `Special` *inheritsinheritance* methods and data from *base class* `General`:

```
int main() {
  Special special_object;
  special_object.general_method();
```

Data needs to be `protected`, not `private`, to be inheritable.

Constructors

When you run the special case constructor, usually the general case needs to run too. By default the 'default constructor', but:

```
class General {
public:
  General( double x,double y ) {};
};
class Special : public General {
public:
  Special( double x ) : General(x,x+1) {};
};
```

Exercise 1

Take your code where a `Rectangle` was defined from one point, width, and height.

Make a class `Square` that inherits from `Rectangle`. It should have the function `area` defined, inherited from `Rectangle`.

First ask yourself: what should the constructor of a `Square` look like?

Exercise 2

Revisit the `LinearFunction` class. Add methods `slope` and `intercept`.

Now generalize `LinearFunction` to `StraightLine` class. These two are almost the same except for vertical lines. The `slope` and `intercept` do not apply to vertical lines, so design `StraightLine` so that it stores the defining points internally. Let `LinearFunction` inherit.

Overriding methods

- A derived class can inherit a method from the base class.
- A derived class can define a method that the base class does not have.
- A derived class *override* a base class method:

```
class Base {
public:
  virtual f() { ... };
};
class Deriv : public Base {
public:
  virtual f() override { ... };
};
```

Back to prime numbers

Exercise 3

The *Goldbach conjecture* says that every even number, from 4 on, is the sum of two primes $p + q$. Write a program to test this for the even numbers up to 20 million.

Make an outer loop over the even numbers. In each iteration, make a `primesequence` object to generate p values. Then, for each p, make a second `primesequence` object to generate q values, and test with these.

For each even number, print out how it is the sum of two primes. If multiple possibilities exist, only print the first one you find.

to remind you. . .

Exercise 4

Write a class primesequence that contains the members of the structure, and the functions nextprime, isprime. The function nextprime does not need the structure as argument, because the structure members are in the class, and therefore global to that function.

Your main program should look as follows:

```
primesequence sequence;
while (sequence.numberfound<nprimes) {
  int number = sequence.nextprime();
  cout << "Number " << number << " is prime" << endl;
}
```

and to see if you really understand this. . .

Exercise 5

The *Goldbach conjecture* says that every even number $2n$ (starting at 4), is the sum of two primes $p+q$:

$$2n = p + q.$$

Equivalently, every number n is equidistant from two primes. In particular this holds for each prime number:

$$\forall_{p\,\mathrm{prime}}\exists_{q\,\mathrm{prime}}\colon r \equiv p + (p-q) \text{ is prime.}$$

Write a program that tests this. You need two prime number generators, one for the p-sequence and one for the q-sequence. For each p value, when the program finds the q value, print the q, p, r triple and move on to the next p.

Allocate an array where you record all the $p - q$ distances that you found. Print some elementary statistics, for instance: what is the average, do the distances increase or decrease with p?

More

- Multiple inheritance: an X is-a A, but also is-a B.
  This mechanism is somewhat dangerous.
- Virtual base class: you don't actually define a function in the base class, you only say 'any derived class has to define this function'.