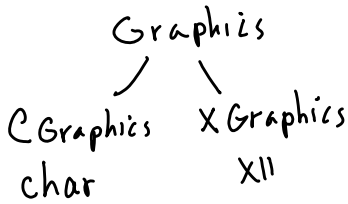


Project Design

- Graphics should have a super-class. Choose sub-class according to capabilities



polymorphism

```

Graphics *g
if (hasX)
    g = new XGraphics
else
    g = new CGraphics
  
```

I'm here to teach you to teach yourself, not to teach you the stuff. Learn to use the basics.

Readings from textbook:

- recursion (chp 6)
- arrays/vectors/strings/templates/STL (chp 7,14,18,22)
- pointers/sorting/big-O/datastructures (chp 8,19,20)
- classes/constructors/friends/inheritance/polymorphism (chp 9-13)
- exceptions (chp 17)
- IO/files/streams/except (chp 15-18)

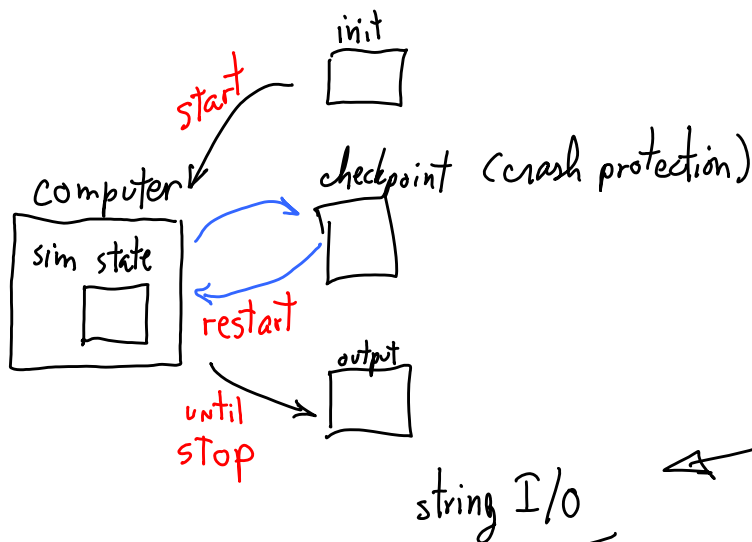
Readings other:

- man/info make
- gdb
- Hall-TinyGL
- Parlante-LinkedListBasics
- Carrano-HeapImplementation
- Wayne-BinaryAndBinomialHeaps

Other issues

Sim: startup, stop, restart, checkpoints

files



Provenance:

- Who (did, has, for)
- Why, When, What (+ files)
- Where it's from/going to
- What's been done to it (exactly!) (can be undone, redone, ...) (by What machine, code, OS)

Headers:

date, condition, state, process, ...

Best: human readable data+header (strictly, not possible)

Worst: encoded, compressed bits. (tradeoffs: performance/security)

const protected from mistaken changes → syntax Corner
r 404-413

int (* const x) deref x and you get
 ↑ x is const

x can point to only one int. after initialization (where?). That int can change value.

const int (* x) def x and you get

x can point to any const int. by re-assignment. Each int is fixed.

- 410 - const method, const object
- 411 - overload method
- 413 - destructor, dyn. mem, when called?
 parent destructor called when?
 " constructor " "
 ↖ which? overridden?

svn info
 which version is this?

diff
 svn diff) differences
 changes

⇒ init list.
 f()
 : 0, 0, 0
 { - }

Head 433 - virtual, string, Template
 (virt string vec)

: g/0 / s/3/5 /
 (ex mode)
 \$
 s/1 /foo/g

Vim

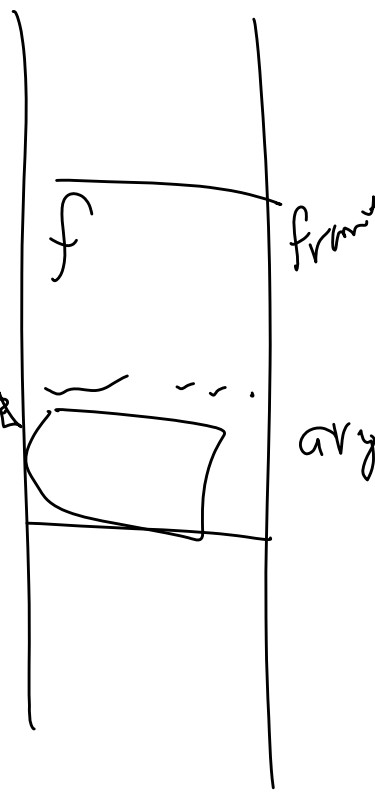
(* . R)

o 0, 0 □
 o . , # d

```

: ., 10 s | ^ | // Node n;
: 'a', b s | ^ | // f(n);

```



```

ptr = new Node;
a = b;
new Node(s);
class Foo {
    Foo(int x(0))
}

```

```

class Bar {
    Bar()
}
foo() {
}
}

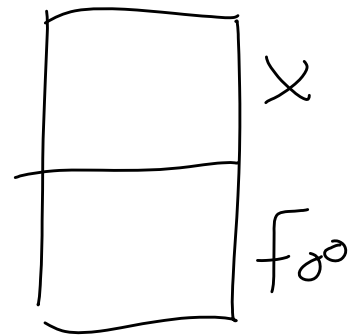
```

= Foo new thisfoo;

b = new Bar();

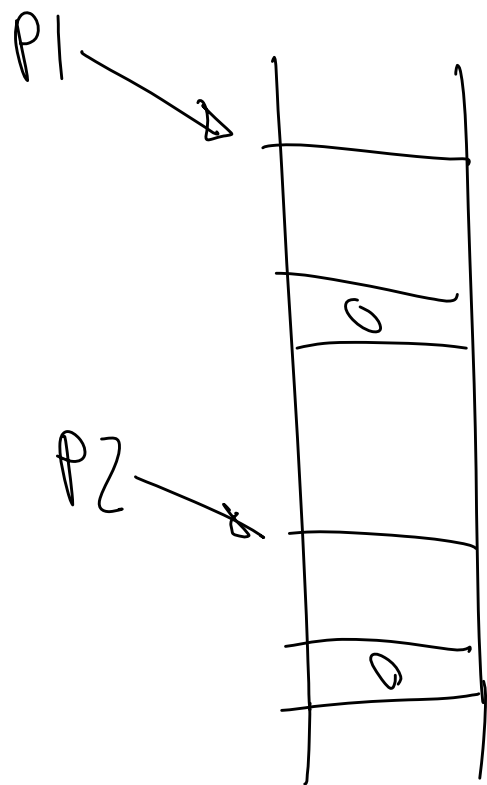
foo: X

X()



foo a;

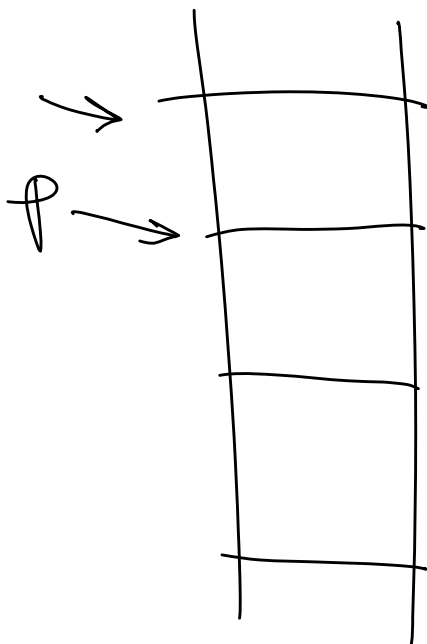
p3 = strcat(p1, p2)



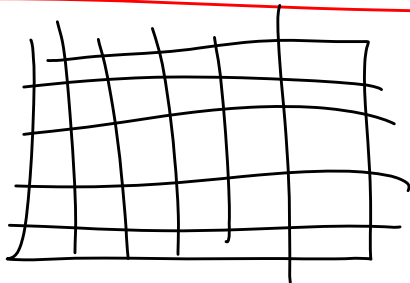
12345 p

12346

p++



fb



→

