CS100J 14 April 2005

Exceptions in Java. Read Chapter 10.

This material will not be tested

Review session Sunday in Olin 155, 1PM-3PM.

Do not miss class or lab on Tuesday. We will start on Matlab in class, and the lab will introduce you to its use on the computer.

I believe it is fundamentally wrong to teach a science like programming by reinforcing the students' intuition when that intuition is inadequate and misguided. On the contrary, our task is to demonstrate that a first intuition is often wrong and to teach the principles, tools, and techniques that will help overcome and change that intuition! Reinforcing inadequate intuitions just compounds the problem. Gries, 1988

Nothing needs changing so much as the habits of others. Mark Twain

On developing programs or algorithms

The majority of the texts for introductory programming courses do not teach you about programming. They simply show you programs. They don't show you thought effective thought processes for the development of programs; They show you programs and expect you to be able to go write your own. Look at other texts to see this for yourself.

My goal is to show you
(1) how to understand programs and
(2) how to develop programs.

Let me make an analogy to make my point clear. Suppose you attend a course in cabinet making. The instructor briefly shows you a saw, a plane, a hammer, and a few other tools, letting you use each one for a few minutes. He next shows you a beautifully-finished cabinet. Finally, he tells you to design and build your own cabinet and bring him the finished product in a few weeks. You would think he was crazy! Gries, D., "What Should We Teach in an Introductory Programming Course", Proc fourth SIGCSE Technical Symp. on Computer Science Education, 1974, pp. 81-89.

On developing programs or algorithms

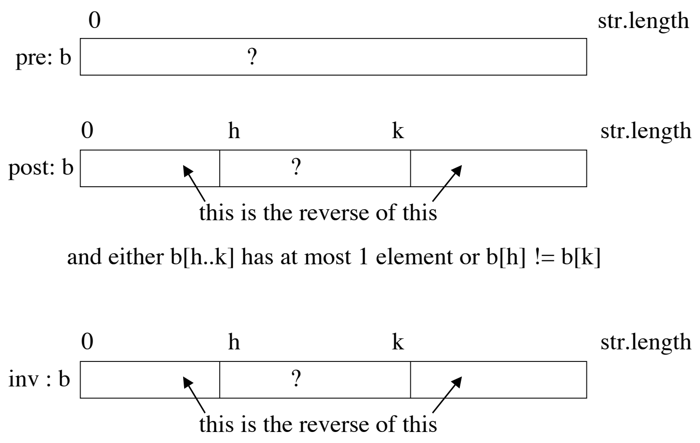
Important points:
1. Managing complexity! (don't let it raise its ugly head)
2. Keep it simple.
3. Correctness should be the driving force!
4. Separate concerns --focus on one thing at a time
5. Use abstraction. E.g. sometimes focus one what rather than how.
6. Whatever you do must be tested thoroughly.

Concern for correctness as a guiding principle for program composition.
Edsger Dijkstra, 1970
http://www.cs.utexas.edu/users/EWD/transcriptions/EWD02xx/EWD288.html

A man, a plan, a canal, Panama! Extension of it, by Dan Huey in 1984:

A man, a plan, a caret, a ban, a myriad, a sum, a lac, a liar, a hoop, a pint, a catalpa, a gas, an oil, a bird, a yell, a vat, a caw, a pax, a wag, a tax, a nay, a ram, a cap, a yam, a gay, a tsar, a wall, a car, a luger, a ward, a bin, a woman, a vassal, a wolf, a tuna, a nit, a pall, a fret, a watt, a bay, a daub, a tan, a cab, a datum, a gall, a hat, a fag, a zap, a say, a jaw, a lay, a wet, a gallop, a tug, a trot, a trap, a tram, a torr, a caper, a top, a tonk, a toll, a ball, a fair, a sax, a minim, a tenor, a bass, a passer, a capital, a rut, an amen, a ted, a cabal, a tang, a sun, an ass, a maw, a sag, a jam, a dam, a sub, a salt, an axon, a sail, an ad, a wadi, a radian, a room, a rood, a rip, a tad, a pariah, a revel, a reel, a reed, a pool, a plug, a pin, a peek, a parabola, a dog, a pat, a cud, a nu, a fan, a pal, a rum, a nod, an eta, a lag, an eel, a batik, a mug, a mot, a nap, a maxim, a mood, a leek, a grub, a gob, a gel, a drab, a citadel, a total, a cedar, a tap, a gag, a rat, a manor, a bar, a gal, a cola, a pap, a yaw, a tab, a raj, a gab, a nag, a pagan, a bag, a jar, a bat, a way, a papa, a local, a gar, a baron, a mat, a rag, a gap, a tar, a decal, a tot, a led, a tic, a bard, a leg, a bog, a burg, a keel, a doom, a mix, a map, an atom, a gum, a kit, a baleen, a gala, a ten, a don, a mural, a pan, a faun, a ducat, a pagoda, a lob, a rap, a keep, a nip, a gulp, a loop, a deer, a leer, a lever, a hair, a pad, a tapir, a door, a moor, an aid, a raid, a wad, an alias, an ox, an atlas, a bus, a madam, a jag, a saw, a mass, an anus, a gnat, a lab, a cadet, an em, a natural, a tip, a caress, a pass, a baronet, a minimax, a sari, a fall, a ballot, a knot, a pot, a rep, a carrot, a mart, a part, a tort, a gut, a poll, a gateway, a law, a jay, a sap, a zag, a fat, a hall, a gamut, a dab, a can, a tabu, a day, a batt, a waterfall, a patina, a nut, a flow, a lass, a van, a mow, a nib, a draw, a regular, a call, a war, a stay, a gam, a yap, a cam, a ray, an ax, a tag, a wax, a paw, a cat, a valley, a drib, a lion, a saga, a plat, a catnip, a pooh, a rail, a calamus, a dairyman, a bater, a canal ---Panama!

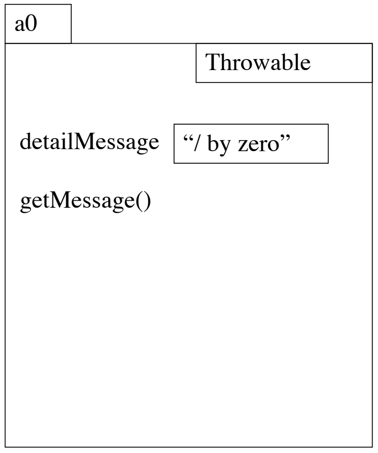
Palindrome: something that reads the same backwards and forwards



and either b[h..k] has at most 1 element or b[h] != b[k]

Exceptions and Errors

In Java, there is a class Throwable:



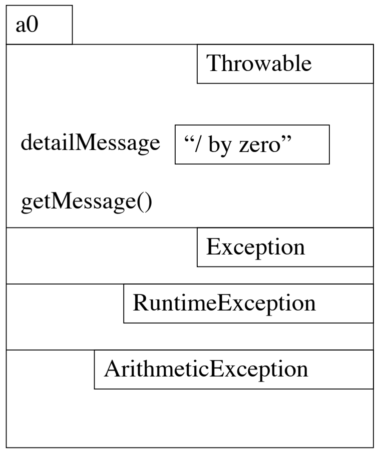
When some kind of error occurs, an exception is "thrown"

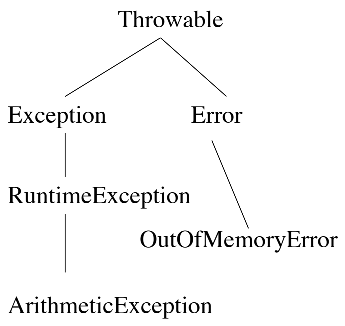
An exception is an instance of class Throwable

(or one of its subclasses)

Exceptions and Errors

So many different kind of exceptions that we have to organize them.





```
/** Parse s as a signed decimal integer and return the integer. The characters
   in the string must all be decimal digits, except that the first character may
   be an ASCII minus sign '-' ('\u002D') to indicate a negative value.
  */

public static int parseInt(String s) throws NumberFormatException {

}
```

Any method may have a throw clause to indicate what it throws. Sometimes, the throws clause is required.