UCLA

CS 31 : Introduction To Computer Science

Howard A. Stahl

Exceptions

- So Far, All Our Classes Have Been Slaves!
  - The Driver Kicked Them Into Doing Something
  - The Class Responded By Running A Method
- But…

Exceptions

- So Far, All Our Classes Have Been Slaves!
  - The Driver Kicked Them Into Doing Something
  - The Class Responded By Running A Method
- Sometimes, Things Go Wrong, Very Wrong
  - The Driver Doesn't Read The Documentation And Sends Bad Parameters
  - Data Files Don't Exist
  - Computers Crash…
  - The List Can Go On And On…

Exceptions

- An Example:
  - Suppose Someone Tries To Create A PowerballTicket With 1,1,1,1,1,1
  - Suppose Someone Tries To `.acceptCard( Card )` More Than 12 Times In A BlackJack Player
  - Suppose Someone Tries To Play Music On Their iPod After The Power Runs Out

Exceptions

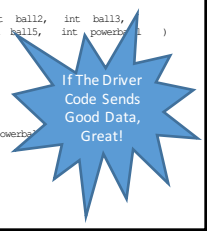
- An Example:

```
PowerballTicket::PowerballTicket(   int  ball1,   int  ball2,   int  ball3,
                                    int  ball4,   int  ball5,   int  powerball   )
{
  if  (ball1   > 0 && ball1   < 69)    mBall1   = ball1;
  if  (ball2   > 0 && ball2   < 69)    mBall2   = ball2;
  if  (ball3   > 0 && ball3   < 69)    mBall3   = ball3;
  if  (ball4   > 0 && ball4   < 69)    mBall4   = ball4;
  if  (ball5   > 0 && ball5   < 69)    mBall5   = ball5;
  if  (ball6   > 0 && ball6   < 27)    mPowerball   = powerball;
}
```

Exceptions

- An Example:

```
PowerballTicket::PowerballTicket(   int  ball1,   int  ball2,   int  ball3,
                                    int  ball4,   int  ball5,   int  powerball   )
{
  if  (ball1   > 0 && ball1   < 69)    mBall1   = ball1;
  if  (ball2   > 0 && ball2   < 69)    mBall2   = ball2;
  if  (ball3   > 0 && ball3   < 69)    mBall3   = ball3;
  if  (ball4   > 0 && ball4   < 69)    mBall4   = ball4;
  if  (ball5   > 0 && ball5   < 69)    mBall5   = ball5;
  if  (ball6   > 0 && ball6   < 27)    mPowerball   = powerball;
}
```



Exceptions

- An Example:

```
PowerballTicket::PowerballTicket(   int  ball1,   int  ball2,   int  ball3,
                                    int  ball4,   int  ball5,   int  powerball   )

  if (ball1   > 0 && ball1   < 69)    mBall1   = ball1;
  if (ball2   > 0 && ball2   < 69)    mBall2   = ball2;
  if (ball3   > 0 && ball3   < 69)    mBall3   = ball3;
  if (ball4   > 0 && ball4   < 69)    mBall4   = ball4;
  if (ball5   > 0 && ball5   < 69)    mBall5   = ball5;
  if (ball6   > 0 && ball6   < 27)    mPowerball   = powerball;
```

But What If It Doesn't…

If The Driver Code Sends Good Data, Great!

Exceptions

- An Example:

```
PowerballTicket::PowerballTicket(   int  ball1,   int  ball2,   int  ball3,
                                    int  ball4,   int  ball5,   int  powerball   )
{
  if (ball1   > 0 && ball1   < 69)    mBall1   = ball1;
  else
      cout   << "ball1   was bad!"   << endl;
  if (ball2   > 0 && ball2   < 69)    mBall2   = ball2;
  else
      cout   << "ball2   was bad!"   << endl;
  if (ball3   > 0 && ball3   < 69)    mBall3   = ball3;
  else
      cout   << "ball3   was bad!"   << endl;
  if (ball4   > 0 && ball4   < 69)    mBall4   = ball4;
  else
      cout   << "ball4   was bad!"   << endl;
```

Exceptions

- An Example:

```
PowerballTicket::PowerballTicket(   int  ball1,   int  ball2,   int  ball3,
                                    int  ball4,   int  ball5,   int  powerball   )
{
  if (ball1   > 0 && ball1   < 69)    mBall1   = ball1;
  else
      cout   << "ball1   was bad!"   << endl;
  if (ball2   > 0 && ball2   < 69)    mBall2   = ball2;
  else
      cout   << "ball2   was bad!"   << endl;
  if (ball3   > 0 && ball3   < 69)    mBall3   = ball3;
  else
      cout   << "ball3   was bad!"   << endl;
  if (ball4   > 0 && ball4   < 69)    mBall4   = ball4;
  else
      cout   << "ball4   was bad!"   << endl;
```

Better Than Nothing, But Not Very Good…

Exceptions

- The Major Problem Is That The Code Continues Uninterrupted Even When It Shouldn't!
  - If The Driver Doesn't Follow The Rules Or Sends Garbage, Code Should Bark Back, Not Pollute The Driver With Bad Data…
- Let's See How Exceptions Help Us Do This…

Revisiting Function Calls

- Functions Call And Return

Caller's Thread

Revisiting Function Calls

- Functions Call And Return

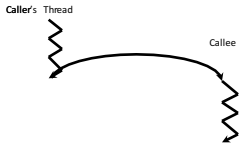
Caller's Thread

Revisiting Function Calls

- Functions Call And Return

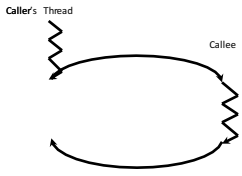
Revisiting Function Calls

- Functions Call And Return



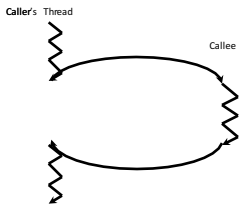
Revisiting Function Calls

- Functions Call And Return



Revisiting Function Calls

- Functions Call And Return



Revisiting Function Calls

- Functions Call And Return

Revisiting Function Calls

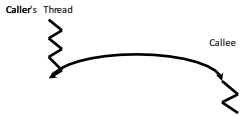
- Functions Call And Return

Revisiting Function Calls

- Functions Call And Return

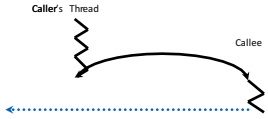
Revisiting Function Calls

- Functions Call And Return



Revisiting Function Calls

- Functions Call And Return

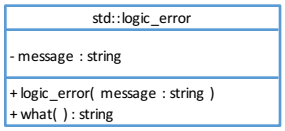


An Exception Is An Alternative Return From A Function That Designates Error

Unless Properly Handled By The Caller, The Program Will End!

Introducing std::logic_error

- A Class That Represents An Error
- Remember To `#include <stdexcept>`



Throwing Exceptions

- To Send An Exception, Use The `throw` Statement
  - `throw(std::logic_error("bad news"));`
- Like A `return` Statement, You Pass A Value Back To The Caller
- `std::logic_error` Is A Class
- Be Sure To `#include <stdexcept>` To Use It

Exceptions

- An Example:

```
PowerballTicket::PowerballTicket(   int  ball1,   int  ball2,   int  ball3,
                                    int  ball4,   int  ball5,   int  powerball   )
{
  if  (ball1   > 0 && ball1   < 69)    mBall1  = ball1;
  else
  {
     std::logic_error   e( "ball1   was  bad!"  );
     throw(   e );
  }
  if  (ball2   > 0 && ball2   < 69)    mBall2  = ball2;
  else
  {
     std::logic_error   e( "ball2   was  bad!"  );
     throw(   e );
  }
```

Exceptions

- Driver Code Would Need To Say:

```
try {
  PowerballTicket t( 1, 1, 1, 1, 1, 1 );
  cout << "This line is never run because an exception happened!" << endl;
  PowerballLottery l;
  l.printWhatHappened( t );
} catch( std::logic_error e ) {
  cout << "ticket that you tried to make failed to be created!" << endl;
}
```

Catching Exceptions

- If The Driver Doesn't Catch An Exception, The Running Program Will End
- The Driver Uses The `try { } catch( )` block To Do Exception Handling
  - You Can Have As Many `catch` Statements As You Like
  - They Are Tried In The Order Listed

Exceptions

- Very Typically, Class Code Throws And Driver Code Catches
  - Classes Typically Throw Exceptions To Designate Failure
  - Driver Code Typically Catches Exceptions

Exception Handling

- The Point Of Exception Handling
- Get Away From All The Garbagey Return-Value Checking That Exists In A C Approach
- Think Positively!
- Put All The Error Handling In One Place