* An exception is an error condition that changed the normal flow of control in a program.
* Exceptions in Java separates error handling from main business logic
* Based on ideas developed in Ada, Eiffel and C++
* Uniform approach for error approach
  + From very unusually
    - out of memory
  + More common ones that your program should check for
    - Out of bounds
  + Java run time system errors
    - Divide by 0
  + Errors that programmer detect and raise deliberately.
* An error can throw and exception
  + Throw <exception object>

Handle

* Prevent exception from happening
* Catch it in the method in which it occurs, and either
  + Fix up the problem and resume normal execution
  + Rethrow it
  + Throw a different exception
* Declare that the method throws and exception
* Note switch
* Checked and unchecked
  + RuntimeException, Error, and their subclass are known as unchecked exceptions
  + All other exceptions are known as checked exceptions, meaning the complier forces the programmer to check and deal with the exceptions
* Error
  + Unchecked
  + Serious system problems
  + Unlikely the program can recover
* RuntimeException
  + Unchecked
  + Logic issue

Catching

* + Try

{

//statement that could cause and exception

}

catch(type)

{

//statements that could handle the exception

}

finally  
{

//Final statements

}

Declaring Exception type

* Inherit from existing exception type
* Provide a default constructor
* And a constructor with one arg, of type String
* Both should call super(astring)

Exception handling required for all read methods

* Also many other system methods

If you use one of Javas built in class methods and it throws an exception, you must catch it (surround with a try/catch block) or rethrow it, or you will get a compile time error