

Object Oriented Software Development Image: Comparison of the second second

Writing high quality software



- Programs should handle unexpected error conditions in a controlled way
 - Report or log error as appropriate
 - Don't "crash"

au

GCU

- Programs should be thoroughly tested
 - Prove that requirements are met
 - Program can run without error but still produce incorrect behaviour
 - Continuous testing of "units" of code and integration of units throughout development

Support for the programmer



- Exception handling in programming language provides a way to deal with error conditions
- Unit testing tools provide support for continuous testing
- Debugging tools provide support for tracing through code as it runs to diagnose error conditions or incorrect behaviour

Run-time errors

(ccu

GCU



- *"Anything that can go wrong, will go wrong"* (Murphy's law)
- Many things can, and do, go wrong when a computer program runs, for example when it:
 - attempts to open a file that doesn't exist
 - tries to convert a string to a numeric type
 - performs a calculation which divides by zero
 - tries to send a message to a null object reference
- If error is not handled, program will crash

What are exceptions?



- In C# (and other languages) when a run-time error occurs an **exception** is generated
- Usually say exception is thrown
- An exception is an object that contains information about a run-time error
- Exception-handling involves writing code which is executed when an exception is thrown
- Interrupts normal flow of execution

<section-header><image><image><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><image><image><image>











System resources

a



- Should be closed when no longer needed
- Exception occurring can prevent code which closes resources being reached
- · Resources not released back to system
- Files can become unavailable, system can become unstable, etc.
- Need to clean up resources after exceptions

菌 **Cleaning up afterwards** StreamWriter sw = null; try (FileStream fs = File.Open(filename, FileMode.Open); sw = new StreamWriter(fs); sw.Write(data); sw.Close(); could get exception while writing data even of file opens successfully catch (FileNotFoundException fnfex) close file normally - this code might not be reached finally if (sw != null) finally block will always execute sw.Close(); need to declare resource outside try block so that it is in scope in finally block Gau

Should we catch exception immediately?

au



菌。

- Wrapping code in try-catch means exception is handled immediately
- This will prevent program crashing, but may not be the most useful behaviour
- For example, does the user need to know about the error?
- · We don't necessarily want to suppress error
- Can opt to handle exception elsewhere, e.g. in user interface code

Exceptions in class library code



- Class library code may be used by many different programs
- Each program may have its own way of making users aware of error situations or of logging errors
- Class library code should not usually handle exceptions completely
- Should pass them onto program to handle

GCU

au

Options for handling an exception



- · Handle the exception condition immediately
- Pass on the unhandled exception to the calling class
- Handle and rethrow the exception to be handled by the calling class
- Throw a new exception to be handled by the calling class





Throws exception	Method where error occurred Looking for anonomiate	
Forwards exception	Method without an exception handler Looking for	
Catches some //	Method with an exception handler	
	main	





Cleaning up (again)

GCU



- Difficult for calling class to clean up resources used by called method
- Can use try-finally in called method
- Doesn't catch exception, but finally block will always execute even if exception occurs
- More convenient shorthand for this in C# with using key word
- Other languages have similar constructs







T

Throwing your own exceptions

- You can use the exceptions mechanism to indicate and handle error conditions which don't crash the program, but do break your program's business rules
- For example:

au

- Invalid password entered
- Attempt to withdraw more than specified limit from a bank account

A custom exception class

œ



8. Exce

- Exception class to be used by a bank account class
- OverdrawnException class, derived from Exception
- Business rule is that customer cannot withdraw an amount which would leave a negative balance in the account
- Exception is used to indicate a situation where this rule has been violated















Testing



 Software must be comprehensively tested to check that it meets the specified requirements and works correctly

- Different types of testing meet different purposes as development proceeds, e.g.:
 - Integration testing as major components are put together
 - System testing against specification
 - Alpha/Beta testing before release by members of target user community
 - Unit testing

Gau

Unit testing



- The process of testing individual classes is known as **unit testing**
- Continuous process during, and even before, development
- Test-Driven Development process involves writing tests first, then developing code until tests are passed
- There is support for unit testing in most modern languages

Unit testing

GCU

au



- Tests should be written to exercise the significant behaviour of all classes
- Unit tests can be repeated as we continue to develop each class and integrate classes
- Make sure that we don't inadvertently break a part of a class which was working correctly
- Make sure changing one class does not break another class which depends on it
- This is known as regression testing







Test class

GCU



- By default, has a name based on name of class to be tested
- Order -> OrderTest
- Each test is defined by a test method
- Basic code for test methods is generated, but needs to be edited to create meaningful tests
- Success/failure defined with Assert statements









au



- Situations where a program does not behave as expected are known as bugs
- Can be errors which halt the program or simply incorrect behaviour
- All forms of testing can discover bugs
- Most bugs should be discovered before release but fixing bugs is also part of postrelease maintenance

Bug-tracking



Teams often use bug-tracking systems

- Testers can report bugs they discover
- Each bug is assigned to a particular developer to resolve
- Developer records resolution of each bug
- Often used with version control
 - Central copy of code, each developer "checks out" a part to work on and "checks in" new code
 - All versions kept, can roll back and define
 numbered builds and release versions

Debugging

GCU

au



- Testing can find bugs, but doesn't fix them
- Developer must examine code to find source of error or test failure
- **Debugger** is a tool which allows you to break program execution at a specific point, step through code line-by-line and watch the values of variables which may help diagnose the cause of the problem
- Most IDEs include debugging tools

<text><list-item><list-item><list-item><list-item><code-block><code-block></code></code>



Debugger options



- Continue go on to next breakpoint if any more are set
- Stop Debugging end execution immediately
- Step Into go to next statement, if statement is method call then go to first line of method
- Step Over go to next statement, do not step into a called method
- Step Out run to end of current method and stop at next statement of calling method

Get







Debugging in practice

(au



- Need to be able to understand the program
- Need to identify most useful place in program to set breakpoints

凿。

8. Excep

• Need to identify critical variables to watch

Object Oriented Software De

 If bug symptom is an error, experience of common exceptions and error messages can help identify likely causes

What's next?	、難、
 You now know how to write robus classes to implement the model f You will go on learn how to creat user interface which allows peop with the system 	st C# for a system. e a graphical le to interact
Object Oriented Software Development	8. Exceptions, testing and debugging 44