Chapter 15

Exception Handling

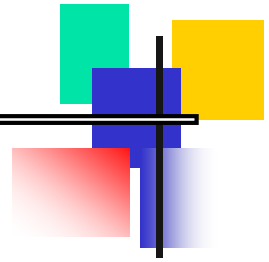
OBJECTIVES

After studying this chapter you will be able to:

- **Use the C++ exception handler.**
- **Use a *throw* statement to raise an exception.**
- **Use the *try* and *catch* statements to handle exceptions.**
- **Use a generic *catch* statement to catch any type of exception.**
- **Create an object to throw exceptions.**
- **Limit the types of exceptions that can be caught.**
- **Modify the default behavior of *unexpected* function.**
- **Modify the behavior of *terminate* function in a program.**
- **Use the C++ *exception* classes.**

HANDLING ERRORS

Figure 15-1 The try and catch statements



```
try
  {
    Code that contains logic to throw an exception
  } // try
catch (error type)
  {
    Exception handler
  } // catch
```

Figure 15-2 Throwing an exception in a separate function

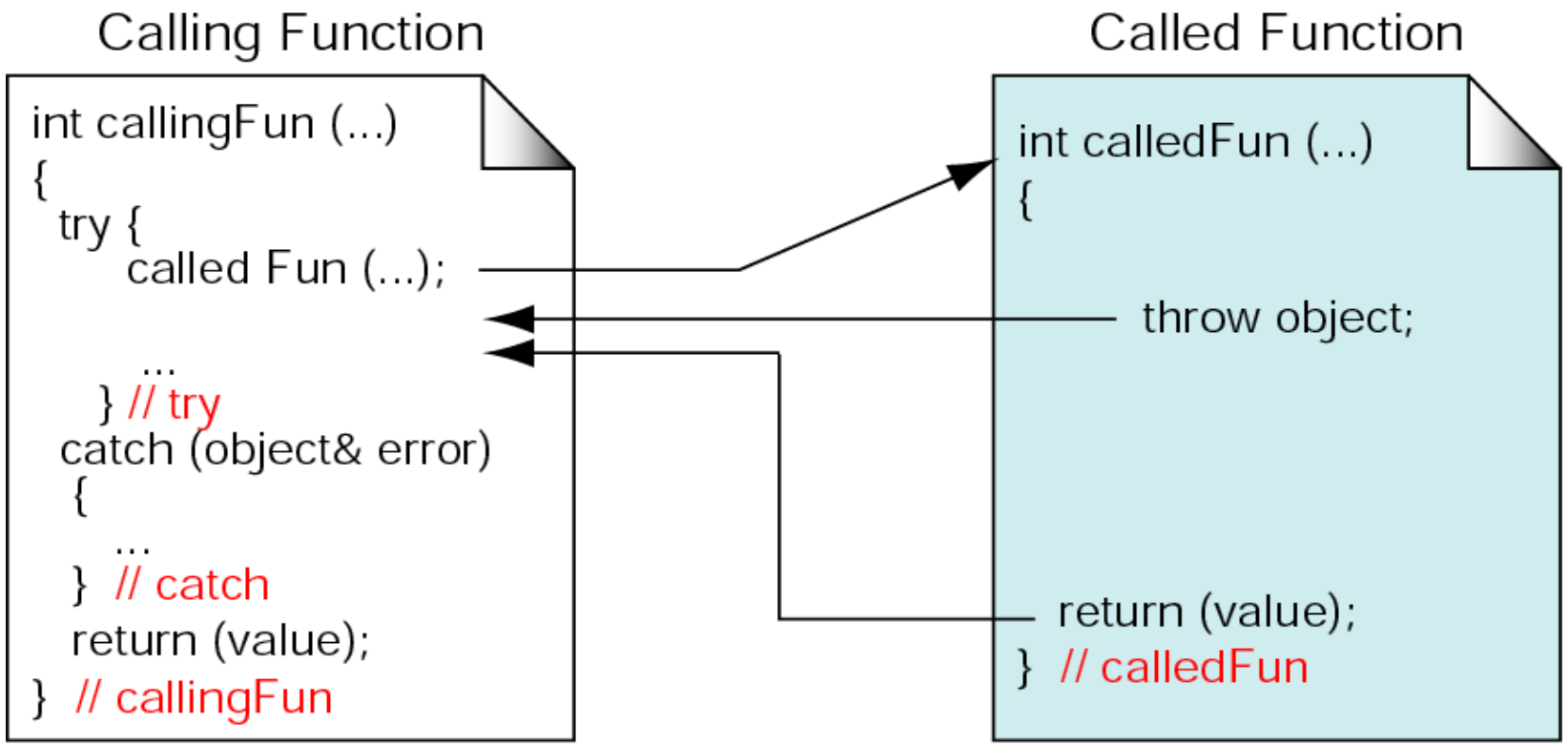


Figure 15-3 Re-throwing an exception

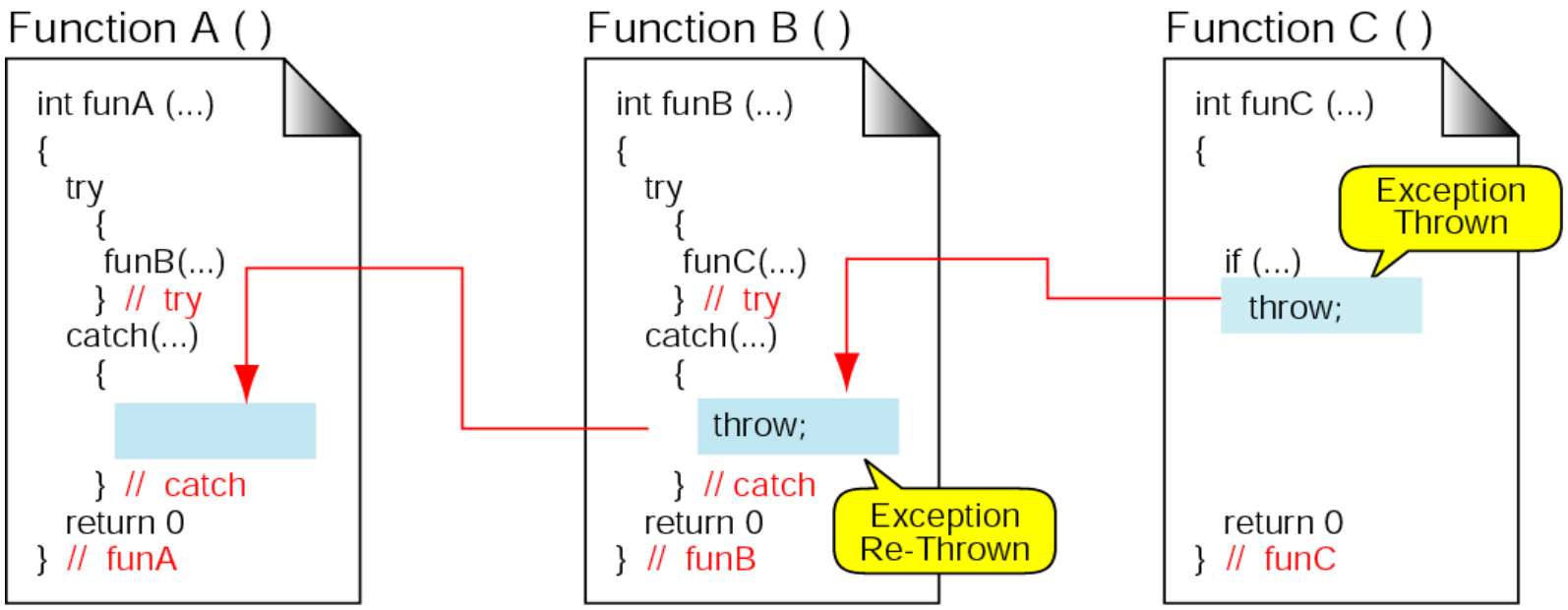
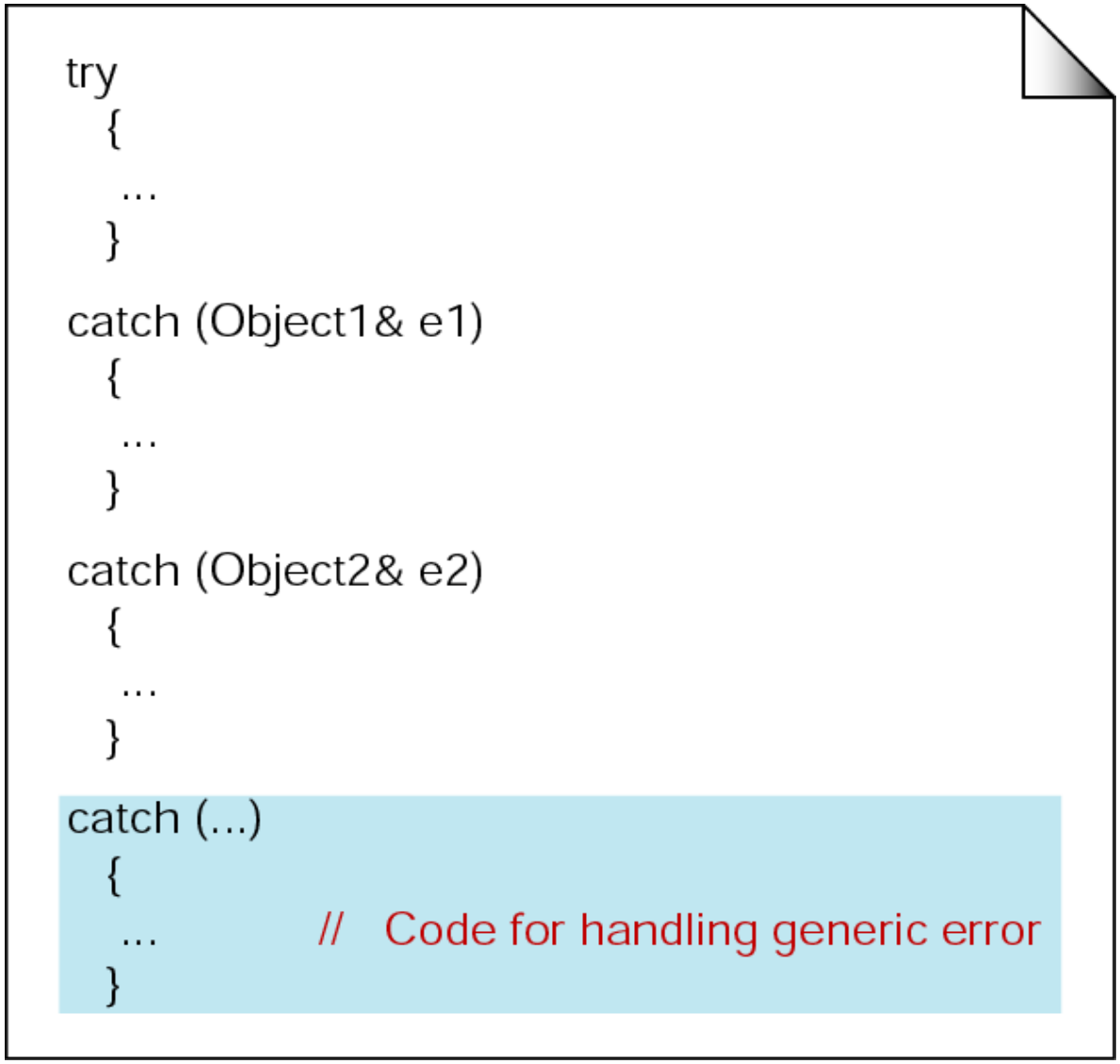


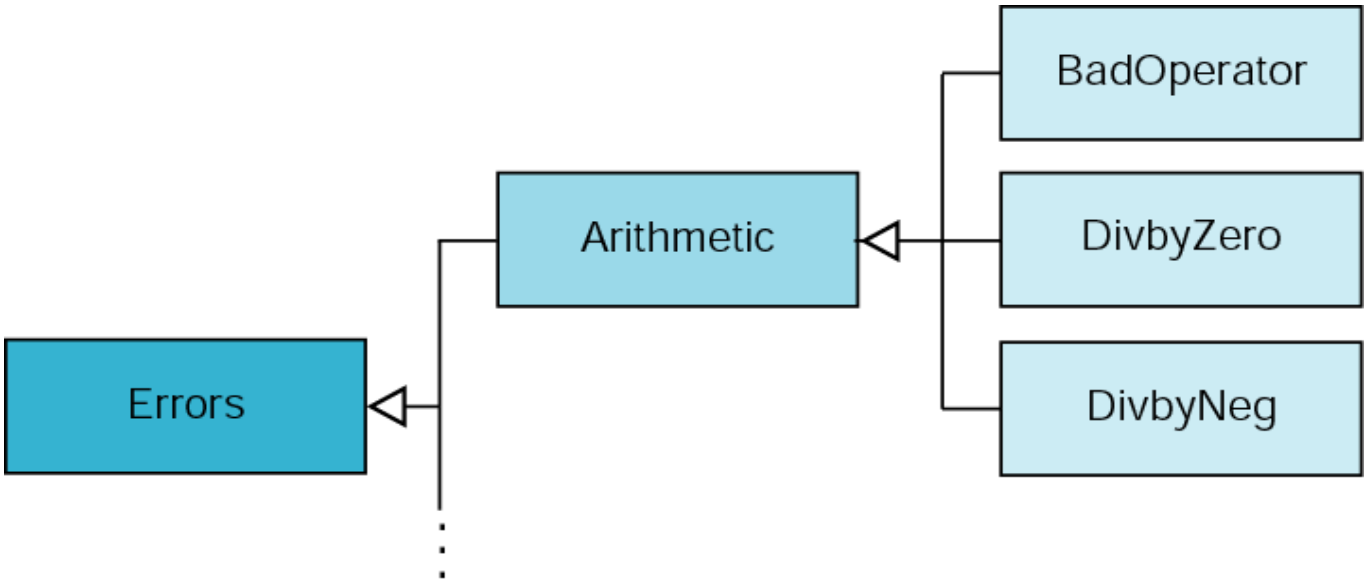
Figure 15-4 Generic exception handler



```
try
  {
   ...
  }

catch (Object1& e1)
  {
   ...
  }

catch (Object2& e2)
  {
   ...
  }

catch (...)
  {
   ...          //  Code for handling generic error
  }
```

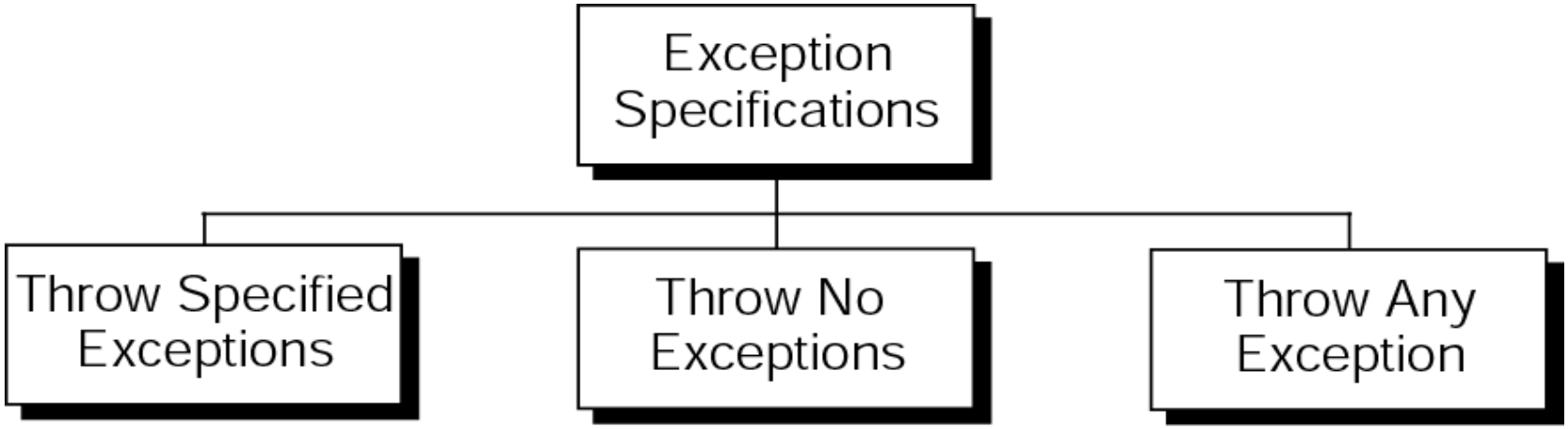
EXCEPTION HANDLING CLASSES

Figure 15-5 Error class design



EXCEPTION SPECIFICATION

Figure 15-6 Exception specifications



EXCEPTIONS IN CLASSES

STANDARD EXCEPTIONS

Figure 15-7 Standard exceptions

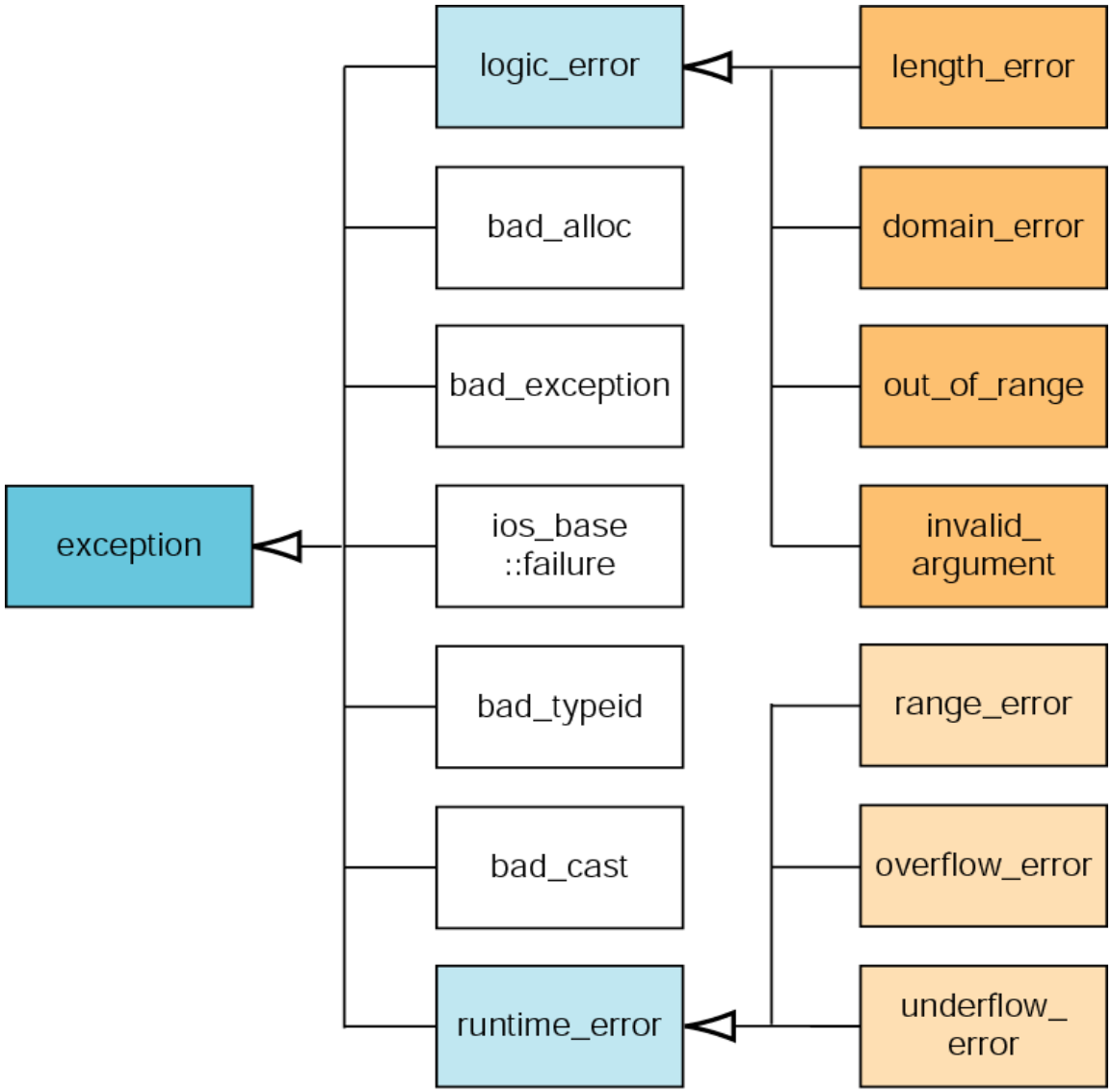
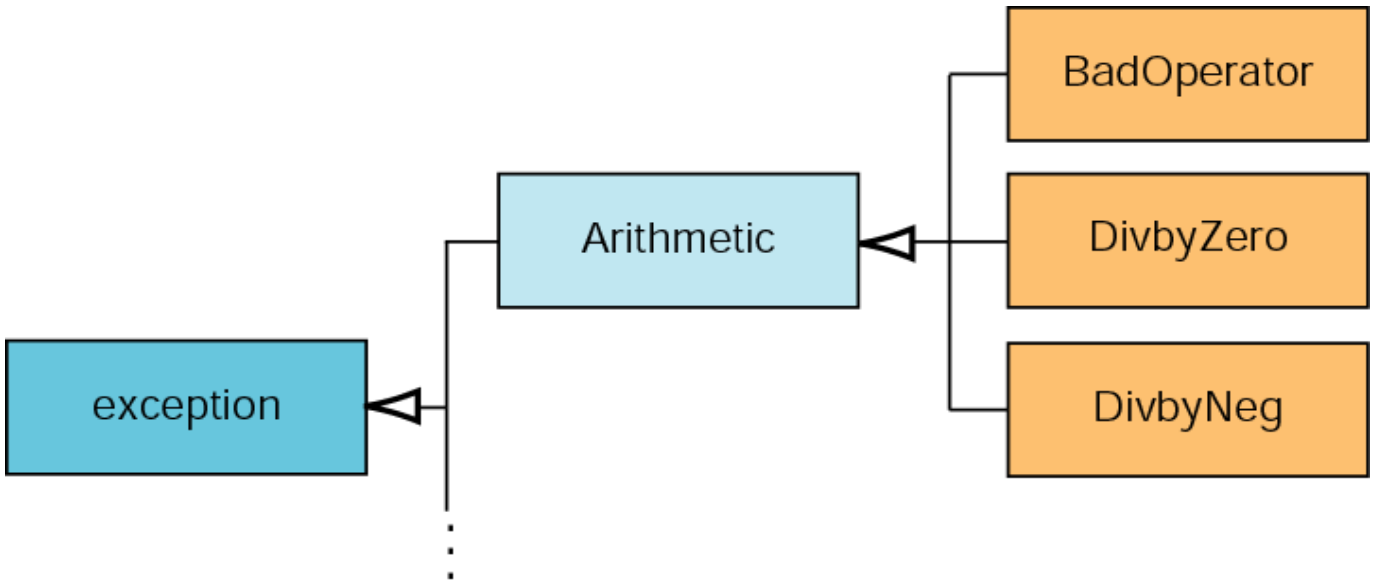


Figure 15-8 Adding errors to standard error class



SOFTWARE ENGINEERING AND PROGRAMMING STYLE