**Code Kata-GCC**

**2019-04-15**

**Prerequisites**

1. Download and install MinGW (e.g., D:\MinGW) [Minimum GNU for Windows]
2. MinGW Wikipedia page: <https://en.wikipedia.org/wiki/MinGW>
3. MinGW: <http://mingw.org/>
4. Add to PATH: D:\MinGW\bin; D:\MinGW\mingw32\bin; D:\MinGW\msys\1.0\bin
5. Download and install Sublime Text (e.g., D:\Sublime Text 3)
6. Sublime Text 3: <https://www.sublimetext.com/3>
7. googletest Primer:
8. <https://github.com/google/googletest/blob/master/googletest/docs/primer.md>
9. googletest release:
10. <https://github.com/google/googletest/releases>
11. Take latest release zipfile, and unzip contents into D:\googletest
12. cd /d/googletest/googletest/make
13. Copy Makefile to Makefile-orig; edit Makefile and comment out references to pthread
14. make
15. ./sample1\_unittest.exe, ./sample2\_unittest.exe, ./sample3\_unittest.exe, ...
16. Make sure sample test suites pass
17. cd /d/googletest/googlemock/make
18. Copy Makefile to Makefile-orig; edit Makefile and comment out references to pthread
19. make
20. ./gmock\_test.exe
21. Make sure gmock test suite pass

**Bowling Game**

1. Unzip bowling\_game-gcc-1stTest.zip under D:\workspace\github.com\$(UserId)\gcc
2. Use editor (e.g., SublimeText 3) and open:
   1. $(BOWLING\_GAME)/make/Makefile (e.g,. Makefile for gtest and test suite for Game class)
   2. $(BOWLING\_GAME)/src/test/unit/bowling\_game\_unittest.cc (e.g., Test suite for Game class)
   3. $(BOWLING\_GAME)/include/bowling\_game.h (e.g., Definition of Game class under test)
   4. $(BOWLING\_GAME)/src/app/bowling\_game.cc (e.g., Implementation of Game class under test)
3. cd /d/workspace/github.com/$(UserId)/gcc/bowling\_game
4. cd make
5. make clean
6. make
7. ./bowling\_game\_unittest.exe
8. Make sure bowling game test suite pass

**Googletest Primer**

Link: <https://github.com/google/googletest/blob/master/googletest/docs/primer.md>

**[玩转Google开源C++单元测试框架Google Test系列(gtest)(总)](https://www.cnblogs.com/coderzh/archive/2009/04/06/1426755.html)**

Link: <http://www.cnblogs.com/coderzh/archive/2009/04/06/1426755.html>

下面是该系列的目录：

[1.玩转Google开源C++单元测试框架Google Test系列(gtest)之一 - 初识gtest](http://www.cnblogs.com/coderzh/archive/2009/03/31/1426758.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

[2.玩转Google开源C++单元测试框架Google Test系列(gtest)之二 - 断言](http://www.cnblogs.com/coderzh/archive/2009/04/06/1430364.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

[3.玩转Google开源C++单元测试框架Google Test系列(gtest)之三 - 事件机制](http://www.cnblogs.com/coderzh/archive/2009/04/06/1430396.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

[4.玩转Google开源C++单元测试框架Google Test系列(gtest)之四 - 参数化](http://www.cnblogs.com/coderzh/archive/2009/04/08/1431297.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

[5.玩转Google开源C++单元测试框架Google Test系列(gtest)之五 - 死亡测试](http://www.cnblogs.com/coderzh/archive/2009/04/08/1432043.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

[6.玩转Google开源C++单元测试框架Google Test系列(gtest)之六 - 运行参数](http://www.cnblogs.com/coderzh/archive/2009/04/10/1432789.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

[7.玩转Google开源C++单元测试框架Google Test系列(gtest)之七 - 深入解析gtest](http://www.cnblogs.com/coderzh/archive/2009/04/11/1433744.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

[8.玩转Google开源C++单元测试框架Google Test系列(gtest)之八 - 打造自己的单元测试框架](http://www.cnblogs.com/coderzh/archive/2009/04/12/1434155.html" \t "http://www.cnblogs.com/coderzh/archive/2009/04/06/_blank)

**[编写优美的GTest测试案例](https://www.cnblogs.com/coderzh/archive/2010/01/09/beautiful-testcase.html)**

Link: <http://www.cnblogs.com/coderzh/archive/2010/01/09/beautiful-testcase.html>

1. 案例的层次结构一定要清晰
2. 案例的检查点一定要明确
3. 案例失败时一定要能精确的定位问题
4. 案例执行结果一定要稳定
5. 案例执行的时间一定不能太长
6. 案例一定不能对测试环境造成破坏
7. 案例一定独立，不能与其他案例有先后关系的依赖
8. 案例的命名一定清晰，容易理解

**Optional IDE: VS Code**

1. Download and install Visual Studio Code (e.g., D:\Microsoft VS Code)
2. Visual Studio Code: <https://code.visualstudio.com/>