Steven Chu

905-094-800

CS 31 Project 5 Report

1. Because this program was a very straightforward and simple example of object oriented programming, I did not notice any significant challenges in the process of implementing it other than needing to do some quick searches on class and method declaration syntax. I had one issue with misinterpreting the rules of the game, and I didn’t realize the order of the balls did not matter, and that the balls can match from ticket to lottery in any order as long as the same values are present.
2. The getter methods can be tested through the series of asserts listed below:

MegaMillionsTicket ticket( 1, 2, 3, 4, 5, 6 );

assert( ticket.getBall1() == 1);

assert( ticket.getBall2() == 2);

assert( ticket.getBall3() == 3);

assert( ticket.getBall4() == 4);

assert( ticket.getBall5() == 5);

assert( ticket.getMegaBall() == 6);

MegaMillionsLottery lottery( 1, 2, 3, 4, 5, 6 );

assert( lottery.getBall1() == 1);

assert( lottery.getBall2() == 2);

assert( lottery.getBall3() == 3);

assert( lottery.getBall4() == 4);

assert( lottery.getBall5() == 5);

assert( lottery.getMegaBall() == 6);

The checkTicket and whatHappened methods can be tested by creating varying ticket objects and using the assert function.

For example:

assert( lottery.checkTicket(ticket) == MegaMillionsLottery::WinningPossibility::FIVEPLUSMEGABALL);

assert( lottery.whatHappened( ticket ) == "You matched 5 balls plus the megaball!" );

ticket = MegaMillionsTicket( 1, 2, 3, 4, 5, 12 );

assert( lottery.checkTicket(ticket) == MegaMillionsLottery::WinningPossibility::FIVE);

assert( lottery.whatHappened( ticket ) == "You matched 5 balls!" );

ticket = MegaMillionsTicket( 1, 2, 3, 4, 15, 12 );

assert(lottery.whatHappened(ticket) == "You matched 4 balls!");

assert(lottery.checkTicket(ticket) == MegaMillionsLottery::WinningPossibility::FOUR);

ticket = MegaMillionsTicket(1,2,3,4,9,6);

assert(lottery.whatHappened(ticket) == "You matched 4 balls plus the megaball!");

assert(lottery.checkTicket(ticket) == MegaMillionsLottery::WinningPossibility::FOURPLUSMEGABALL);

This segment tests several possible outcomes, and the process can be repeated for every listing in the WinningPossibility enumeration.