Ayan Nath

605910299

**Obstacles**

* One of the main obstacles that I overcame was trying to make the h files work with the cpp files flawlessly. For example, since I was declaring the methods in the h file and implementing them in the cpp file, most times I would often forget to put the correct parameters in the actual cpp file. Additionally, I would sometimes incorrectly add the public and private words in the cpp file when they should be in the h file. Overall, however, the obstacles for this project were not that difficult to overcome compared to the other projects. Rather, the main difficulty in this project was making sure all the files worked together flawlessly.

**Test Cases**

**MovieTicket methods:**

MovieTicket t( "Batman", MovieTicket::Day::MONDAY, MovieTicket::Time::TWOPM, MovieTicket::Kind::BOMB, MovieTicket::Type::ADULT ); // tests MovieTicket constructor

assert( t.getMovie() == "Batman" ); // tests getMovie()

assert( t.getDay() == MovieTicket::Day::MONDAY ); //tests getDay()

assert( t.getTime( ) == MovieTicket::Time::TWOPM ); //tests getTime()

assert( t.getKind( ) == MovieTicket::Kind::BOMB ); // tests getKind()

assert( t.getType( ) == MovieTicket::Type::ADULT );//tests getType()

assert( t.isWeekendTicket( ) == false );//tests isWeekendTicket()

assert( t.isWeekdayTicket() == true );// tests isWeekdayTicket()

assert( t.isMatineeTicket( ) == true );//tests isMatineeTicket()

assert( t.isEveningTicket() == false );// tests is EveningTicket()

assert( std::to\_string( b.getCost() ) == "3.5"// tests getCost()

**BoxOffice methods:**

BoxOffice b( "1234 UCLA Drive ", "The Lego Movie" ); // tests BoxOffice constructor

assert( b.getAddress() == "1234 UCLA Drive" ); // tests to see if getAddress() works

assert( b.whatsPlaying() == "The Lego Movie" ); //tests to see if whatsPlaying() works

assert( b.getWeekendTickets( ) == 0 ); // tests to see if getWeekendTickets() starts at 0

assert( b.getWeekdayTickets( ) == 0 );// tests to see if getWeekdayTickets() starts at 0

assert( b.getMatineeTickets( ) == 0 );// tests to see if getMatineeTickets() starts at 0

assert( b.getEveningTickets( ) == 0 );// tests to see if getEveningTickets() starts at 0

assert( std::to\_string( b.getRevenue() ) == "0.000000" ); // tests if getRevenue() starts at 0

MovieTicket m = b.buyTicket( "Jumanji", MovieTicket::Day::MONDAY, MovieTicket::Time::NOON, MovieTicket::Kind::REGULAR, MovieTicket::Type::ADULT ); //this should fail because the requested movie string value must match the BoxOffice's mMovie value. This also tests the buyTicket function

MovieTicket m = b.buyTicket( "The Lego Movie", MovieTicket::Day::MONDAY, MovieTicket::Time::NOON, MovieTicket::Kind::REGULAR, MovieTicket::Type::ADULT ); //this call to buyTicket should not fail because the requested movie string value matches the BoxOffice’s mMovie value.