Tyler Stovsky

705512370

1. One notable obstacle I overcame was when I was writing the freeFlight function for the FrequentFlyerAccount. At first, I tried to change the PlaneFlight class by calling setCost, setMileage, setFromCity, and setToCity individually. This worked well, but failed when trying to create a desired flight that swapped the destinations of the existing flight. To fix this, I learned that I could simply re-call the constructor for the existing flight which allowed me to avoid any issues where fromCity and toCity would be temporarily set to the same value.

Another obstacle I overcame was that I was having trouble with what to #include on which file. I was having issues getting my program to build because I had used #include in places where I never used its contents. To fix this I went to each file and made sure that I used everything I #include’d.

2. PlaneFlight:

Initialize name empty string **PlaneFlight f(“”, “CLE”, “LAX”, 100.00, 2350)**

Initialize fromCity empty string **PlaneFlight f(“Tyler”, “”, “LAX”, 100.00, 2350)**

Initialize toCity empty string **PlaneFlight f(“Tyler”, “CLE”, “”, 100.00, 2350)**

Initialize fromCity to equal toCity **PlaneFlight f(“Tyler”, “CLE”, “CLE”, 100.00, 2350)**

Initialize cost less than zero **PlaneFlight f(“Tyler”, “CLE”, “LAX”, -350.00, 2350)**

Initialize cost equal to zero **PlaneFlight f(“Tyler”, “CLE”, “LAX”, 0.00, 2350)**

Initialize mileage less than zero **PlaneFlight f(“Tyler”, “CLE”, “LAX”, 100.00, -452)**

Initialize mileage equal to zero **PlaneFlight f(“Tyler”, “CLE”, “LAX”, 100.00, 0)**

**Assume this object will be used for test cases below**

Valid initialization **PlaneFlight f(“Tyler”, “CLE”, “LAX”, 100.00, 2350)**

**f.getCost()**

Cost less than zero **f.setCost(-123.45)**

Cost equal to zero **f.setCost(0.00)**

Cost greater than zero **f.setCost(256.00)**

**f.getMileage()**

Mileage less than zero **f.setMileage(-123)**

Mileage equal to zero **f.setMileage(0)**

Mileage greater than zero **f.setMileage(893)**

**f.getName()**

Empty string name **f.setName(“”)**

Valid name **f.setName(“Chris”)**

**f.getFromCity()**

Empty string fromCity **f.setFromCity(“”)**

fromCity set equal to toCity **f.setFromCity(“LAX”)**

Valid fromCity **f.setFromCity(“DEN”)**

**f.getToCity()**

Empty string toCity **f.setToCity(“”)**

toCity set equal to fromCity **f.setToCity(“CLE”)**

Valid toCity **f.setToCity(“DEN”)**

FrequentFlyerAccount:

Initialize name empty string **FrequentFlyerAccount account(“”)**

Valid initialization **FrequentFlyerAccount account(“Tyler”)**

**account.getBalance()**

**account.getName()**

FrequentFlyerAccount account(“Tyler”)

PlaneFlight wrongName(“Bob”, “CLE”, “LAX”, 100.00, 2350)

Names do not match **account.addFlightToAccount(wrongName)**

PlaneFlight freeFlight(“Tyler”, “CLE”, “LAX”, 0.00, 2350)

Flight is free **account.addFlightToAccount(freeFlight)**

PlaneFlight validFlight(“Tyler”, “CLE”, “LAX”, 63.00, 2350)

Valid flight to add **account.addFlightToAccount(validFlight)**

Mileage less than balance **account.canEarnFreeFlight(2000)**

Mileage equal to balance **account.canEarnFreeFlight(2350)**

Mileage greater than balance **account.canEarnFreeFlight(3570)**

Enough balance for a free flight **account.freeFlight(“DEN”, “ORD”, 1003, validFlight)**

Not enough for a free flight **account.freeFlight(“DEN”, “ORD”, 3010, validFlight)**

Mileage equals balance **account.freeFlight(“DEN”, “ORD”, 2350, validFlight)**

Unchanged destination **account.freeFlight(“CLE”, “LAX”, 2350, validFlight)**

Destinations swap **account.freeFlight(“LAX”, “CLE”, 2350, validFlight)**

fromCity empty string **account.freeFlight(“”, “LAX”, 3010, validFlight)**

toCity empty string **account.freeFlight(“CLE”, “”, 3010, validFlight)**