**CS 162 Worksheet 8**

1. **Analyzing common errors from assignment 3:**

**Common mistake 1**: Redeclaring variables in the constructor.

class Flight {

private:

string flight\_num;

string curr\_loc;

string dest;

...

public:

Flight();

...

};

Flight:: Flight () { // constructor

string flight\_num = “A123”;

string curr\_loc = “B”;

string dest = “A”;

...

}

...

Analyze the code above. Would the member variables of a Flight object be initialized after calling the constructor? Why or why not? How would you fix the code?

**Common mistake 2**: Creating extra object(s) when working with class composition.

class Manager {

private:

Airport\* a\_arr;  
 int num\_airports;

...

public:

void populate(); //populate airport(s) detail

void print\_all();

...

};

class Airport {

private:

Flight\* f\_arr;

int num\_flights;

public:

void populate\_airport();

void print\_airport();

...

};

int main () {

Manager m;

int num\_airports = 3;

Airport\* a\_arr = new Airport [num\_airports];

for (int i = 0; i < 3; i++)

a\_arr[i].populate\_airport();

m.print\_all();

...

}

Assuming all member functions are correctly implemented. Are the Airport objects within the Manager m loaded/populated? Why or why not? How would you fix the code?

**Common mistake 3:** A chain of accessor calls.

void Manager::print\_very\_first\_flight() {

cout << “Flight num: ” << a\_arr[0].get\_f\_arr()[0].get\_flight\_num() << endl;

cout << “Current at: ” << a\_arr[0].get\_f\_arr()[0].get\_curr\_loc() << endl;

cout << “Destination: ” << a\_arr[0].get\_f\_arr()[0].get\_dest() << endl;

...

}

In main():

Manager m;

m.print\_very\_first\_flight();

What is the issue with the print\_very\_first\_flight() function above? Why is it a bad idea to use a chain of accessors to get the internals of Flight class from the Manager? How would you fix the code?

**Class Relationship:**

1. Given the following possible classes, list at least three “has-a” relationship, and three “is-a” relationship.

Animal Dog Shape Mammal Triangle Teeth

Vehicle Person Driver Wheel Truck Space Shuttle

**Accessibility:**

1. Explain the difference between public, private, and protected.

**Inheritance:**

Given the following code, discuss the following and write code to prove your answers.

struct Card {

int rank; // 1-13

string suit; // “heart”, “spade”, “diamond”, “club”

};

class Cardgame {

protected:

Card \*deck;

int num\_cards;

public:

Cardgame();

~Cardgame();  
 void play\_game();

};

class Gofish : public Cardgame { //inherited from Cardgame

private:

int max\_players;

public:

Gofish();

~Gofish();

};

...

1. If we create a child object, i.e. Gofish g;
   1. What is inherited and not inherited?
   2. What is accessible and not accessible?
   3. In what order is the Cardgame constructor and Gofish constructor called?
   4. When the object g is out of scope, in what order is the Cardgame destructor and Gofish destructor called?