CMSC162 Intro to Algorithmic Design II Blaheta

Homework 1

Due: 24 September 2014

Problem 1.1

Assume that the `Card` class is defined as in Lab 4, and consider the following not-very-useful code:

```
void func (Card a, Card& b, Card* c)
{
  cout << a.getSuit() << b.getSuit() << c->getSuit() << endl;

  a = Card(9, ’H’);
  b = Card(8, ’S’);
  c = new Card(1, ’H’);

  cout << a.getSuit() << b.getSuit() << c->getSuit() << endl;
}

int main()
{
  Card x (3, ’C’);
  Card* y = new Card (7, ’D’);

  func (x, x, y);
}
```

Its output is what you’d probably expect:

```
CCD
HSH
```

For this problem, though, draw out two diagrams of what’s in memory, and where: one diagram for the state of memory at the time of the first `cout` statement, and another to show the changes as of the second `cout` statement.

Problem 1.2

Consider a class `RpgPc` that has the task of representing players in a role-playing game, each of which has a name and a series of stats:

```
class RpgPc
{
  private:
    string name;
    int* stats;
...
```

The following is proposed for one of its constructors. Some lines that aren't of current interest are omitted, as is the definition of `average` (which computes the average of a six-element array of integers).

```
    RpgPc(string n) : name(n)
    {
      int* statsA = new int[6];
      int* statsB = new int[6];
      /* ... randomly generate values in both arrays ... */
      double avgA = average(statsA);
      double avgB = average(statsB);
      if (avgA > avgB)
        stats = statsA;
      else
        stats = statsB;
    }
```

This introduces a memory leak.

a. Explain why.

b. Draw a diagram of memory at the end of running the constructor that illustrates the problem.

c. Explain how to fix the memory leak.