CSE1502 Fall 2008 Final Exam
Open book, open notes, no computers.

1. What does the following print? (4 points each)

```
string s = "testing";      ANSWERS
cout << s[2];              s

cout << s+"2";            string2

cout << char(s[2]+2);     u

cout << s.substr(2, 3);   sti

cout << int(s.size());    7
```

2. Write a function **min** taking a vector<int> v and returning the smallest element. If v is empty, return 0. For example, if v contains {2, -3, 8}, min(v) would return -3 (20 points).

```
// ONE ANSWER
int min(vector<int> v)
{
  if (int(v.size())==0)
    return 0;
  int result=v[0];
  for (int i=1; i<int(v.size()); ++i)
    if (v[i]<result)
      result=v[i];
  return result;
}
```

3. Write a function **string2map** taking a string and an empty map<char, int> and modifies the map so that each character that occurs at least once in the string is mapped to the number of times it occurs. For example, (20 points).

```
map<char, int> m;
string2map("testing", m);  // sets m
cout << m['t'];  // 2
cout << m['g']; // 1
cout << int(m.size());  // 6
```

```
// ONE ANSWER
void string2map(string s,
                map<char,int>&m)
{
  for (int i=0; i<int(s.size()); ++i)
    ++m[s[i]];
}
```

Name _____

4. Write a function **smallest** taking two doubles and returning the one closest to zero. For example, smallest(-3.5, 4.2) would return -3.5, smallest(-2, 1.5) would return 1.5 (20 points).

```
// ONE ANSWER
double smallest(double a, double b)
{
  if (fabs(a)<fabs(b))
    return a;
  else
    return b;
}
```

5. Class **Distance** represents a distance. It has a constructor that takes a number and either "in" or "ft" to set the distance in inches or feet, and member functions to get the value in either inches or feet. The class definition and an example of its use is shown below. Write the code for the two non-inlined functions. (20 points).

```
class Distance
{
private:
  double in;  // in inches
public:
  Distance(double value, string units);
  double getInches() {return in;}
  double getFeet();
};

int main()
{
  Distance x(6.0, "in"), y(1.25, "ft");
  cout << x.getFeet();  // 0.5
  cout << y.getInches();  // 15.0
  return 0;
}
```

```
// ONE ANSWER
Distance::Distance(double value,
                   string units)
{
  if (units=="in")
    in=value;
  else
    in=value*12;
}

double Distance::getFeet()
{
  return in/12.0;
}
```