CSE2050 Fall 2007 Exam #2. Open book, open notes, no computers. Name ___________________________

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

```
void f(int x)
{
  if (x < 0)
    return -f(-x);
  else if (x > 0)
    return 2 + f(x - 1);
  else
    return 0;
}

int main()
{                  ANSWERS
  cout << f(0);      0
  cout << f(1);      2
  cout << f(-2);     -4
  cout << f(300);    600
}
```

Note: f(x) returns 2*x. I gave a 5 point bonus to anyone who caught my error that f should return int, not void.

Is there any value of **x** that would cause infinite recursion? If so, give an example (4 pts).

ANSWER: No.

2. Write a function **triple** taking an int by reference and returning void. After calling, the argument should have 3 times its original value, for example (20 pts).

```
  int a = 10;
  triple(a);
  cout << a;  // 30
  triple(a);
  cout << a;  // 90
```

```
// ANSWER
void triple(int& x)
{
  x = x * 3;
}
```

3. Write a function **len** that takes a string argument s by value and returns the length of s if all the characters of s are the same, or 0 if any two characters are different. For example (20 pts).

```
  cout << len("aaaaa");// 5
  cout << len("baa");  // 0
  cout << len("x");    // 1
  cout << len("");     // 0
  cout << len("aa") + len("bbbb"); // 6
```

```
// ANSWER
int len(string s)
{
  for (int i=1; i<int(s.size()); ++i)
    if (s[i] != s[0])
      return 0;
  return int(s.size());
}
```

4. An object of class **Employee** is initialized with a name (string) and hourly pay rate (double). It has a member function **print()** which prints this information, and a member function **pay(hours)** which takes the number of hours worked (as a double) and returns the amount to be paid (hours * pay rate, as a double). Write class Employee. All data members should be private. An example of its use is shown. (40 pts).

```
  Employee x("Bob", 9.25);
  x.print();  // Bob earns 9.25 per hour
  double amount = x.pay(40.0);
  cout << amount;  // 370.0
```

```
// ANSWER
class Employee
{
private:
  string name;
  double rate;
public:
  Employee(string n, double r)
  {
    name = n;
    rate = r;
  }
  void print()
  {
    cout << name << " earns " << rate
         << " per hour\n";
  }
  double pay(double hours)
  {
    return rate * hours;
  }
};
```

Note: member functions need not be inlined as shown.