

1. (25 pts) **Short Answer.** Provide brief (1-3 sentence) answers to the following:

- a) When defining a class, why is it considered a good practice to declare class variable as private, instead of public?
  
  
  
  
  
  
  
  
  
  
- b) What is a “memory leak” and why is it undesirable?
  
  
  
  
  
  
  
  
  
  
- c) If we are overloading the + operator for class `Foo`, as in:  
  

```
friend Foo operator+(Foo &f1, Foo &f2);
```

  
Why do we need to use the keyword `friend`?
  
  
  
  
  
  
  
  
  
  
- d) Given the declaration: `int arr[100];`  
This variable is passed into function `foo` as: `foo(arr);`  
The function is defined as: `void foo(int *p);`  
Is there a compiler error as far as the data types are concerned to pass `arr` into the function? Explain why or why not.
  
  
  
  
  
  
  
  
  
  
- e) You have multiple files in your program that all include “`database.h`”. This file contains class definitions for your database. You are getting redefinition errors. How do you solve this problem?

## 2. (18 pts) Find the Bugs

All of the following code snippets have at least one bug or problem. Assume the necessary headers/includes are in place to make a working program.

a) The following code has a compiler error. Identify and correct the error.

```
class Foo
{
private:
    int x;
public:
    Foo();
    void setX(int x) const;
    int getX() const;
};

Foo::Foo() : x(0)
{ }

void Foo::setX(int x) const
{
    this->x = x;
}

int Foo::getX() const
{
    return x;
}
```

b) The following C++ code is rather meaningless, but one line gives a compiler error. Which one and why?

```
int *p1, *p2, arr[5] = {0, 1, 2, 3, 4};
p1 = (arr + 3);
p2 = &p1[1];
arr = (p1 + 2);
```

c) The following function creates a dynamic array, uses it, then exits. Identify and fix the error.

```
void foo()
{
    int *a;
    a = new int[10];
    a[3] = 100;
    cout << a[0] << " " << a[3] << endl;
}
```

**3. (22 pts) Give the output for the following C++ snippets.**

a) Given the C++ declarations:

```
int *p1, *p2, arr[5] = {0, 1, 2, 3, 4} , x=4;
```

1) What is the output for:

```
p1 = &x;  
x = 5;  
cout << *p1 << " " << x << endl;
```

2) What is the output for:

```
x = 10;  
p1 = &x;  
p2 = p1;  
*p1 = 3;  
cout << *p2 << " " << x << endl;
```

3) What is the output for:

```
p1 = arr;  
cout << *p1 << endl;  
p1 +=2;  
cout << *p1 << endl;
```

b) Give the output when the following program runs.

```
#include <iostream>  
using namespace std;  
  
class CSCE211Test  
{  
public:  
    CSCE211Test();  
    ~CSCE211Test();  
    void set(int i);  
    int get();  
private:  
    int x;  
};  
  
CSCE211Test::CSCE211Test ()  
{  
    cout << "Initializing." << endl;  
    x=0;  
}  
  
CSCE211Test::~~CSCE211Test ()  
{  
    cout << "Exiting." << endl;  
}
```

```

void CSCE211Test::set(int i)
{
    x=i;
}

int CSCE211Test::get()
{
    return x;
}

void foo()
{
    CSCE211Test z;
    cout << z.get() << endl;
}

int main()
{
    CSCE211Test t1, t2;

    t1.set(1);
    t2.set(2);
    foo();
    t2 = t1;
    t1.set(10);
    cout << t1.get() << endl;
    cout << t2.get() << endl;
    return 0;
}

```

#### 4. (25 pts) Trick or Treat!

While trick-or-treating you have kept track of the candy received at different houses so that you can plan which houses to visit next Halloween. You have recorded the candy, address, and a candy rating from 1-4 where 4 is the best. Here is a main method with relevant items in bold:

```

int main()
{
    TrickOrTreat housesVisited[5];
    housesVisited[0] = TrickOrTreat("Twix","201 Providence Dr.", 4);
    housesVisited[1] = TrickOrTreat("Snickers","211 Providence Dr.", 3);
    housesVisited[2] = TrickOrTreat("Granola Bar","34 Infinite Loop", 1);
    // Note: housesVisited[3] is not set
    housesVisited[4] = TrickOrTreat("Tootsie Roll","505 Lake Otis", 2);

    cout << "These houses had good candy" << endl;
    for (int i = 0; i < 5; i++)
    {
        if (housesVisited[i].getRating() >= 3)
            cout << housesVisited[i].getCandy() << " " <<
                housesVisited[i].getAddress() << endl;
    }
    cout << endl;
}

```

(continues on next page)

```

cout << "These houses should be avoided next year" << endl;
for (int i = 0; i < 5; i++)
{
    if (housesVisited[i].getRating() == 1)
        cout << housesVisited[i].getCandy() << " " <<
            housesVisited[i].getAddress() << endl;
}
cout << endl;

cout << "These indices were not set to a TrickOrTreat object" <<
    endl;
for (int i = 0; i < 5; i++)
{
    if (housesVisited[i].getRating() == 0)
        cout << i << endl;
}
}

```

The output when this program runs should be:

```

    These houses had good candy
    Twix 201 Providence Dr.
    Snickers 211 Providence Dr.

    These houses should be avoided next year
    Granola Bar 34 Infinite Loop

    These indices were not set to a TrickOrTreat object
    3

```

Write the `TrickOrTreat` class definition and the implementation of the functions/constructors that would result in the output shown above.

## 5. (10 pts) Sorting an Array of Objects

Listed below is code we covered in class to implement selection sort on an array of integers.

```
void swap(int &a, int &b)
{
    int temp = a;
    a = b;
    b = temp;
}

void selectionSort(int a[], int n)
{
    for (int i = 0; i < n-1; i++)
    {
        int indexOfMin = i;
        for (int j = i+1; j < n; j++)
        {
            if (a[j] < a[indexOfMin])
                indexOfMin = j;
        }
        swap(a[i], a[indexOfMin]);
    }
}
```

Modify the swap and selection sort code to sort an array of `TrickOrTreat` objects, as described in the previous problem. The objects should be sorted by rating from smallest to largest. A sample call might look like:

```
selectionSort(housesVisited, 5);
```