Review Activity 3

Using Pointers to Access Memory Locations

1) What will be the output of the following code fragment? Explain.

```
int* p = new int(32);
int* q = new int(24);
*p = *q;
*q = 42;
cout << *p << " and " << *q <<endl;
```

2) If "`cout << *p1 << endl;`" outputs `5`, and
"`cout << *p2 << endl;`" outputs `6`,
what will be the output of "`cout << p1 + p2 << endl;`"? Explain.
`p1` and `p2` are of `int*` type.

3) If "`cout << p3 << endl;`" outputs `0x596834`,
and "`cout << *p4 << endl;`" outputs `1`,
what will be the output of
"`cout << p3 + *p4 << endl; cout << p3 - *p4 << endl;`"? Explain.
`p3` and `p4` are of `int*` type.

4) `int* foo()` that is given below returns the address of a local variable. Can this address be used by the caller to store other values reliably? Explain.

```
int* foo() {
      int a = 5;
      return &a;
}
```

Example use:

```
int *p = foo();
*p = 7;
```

5) What will be the output of the following code fragment? Explain.

```
int *p = new int(56);
int *q = new int(56);
cout << *p << endl;
cout << *q << endl;
delete p;
delete q;
cout << *p << endl;
cout << *q << endl;
```

6) Something "bad" will happen if this code is run. Explain.

```
int i = 66;
int *ip = &i;
delete ip;
```

7) In the following code fragment, where is `t->i` located? Is it on the stack or heap?

```
class Test {
   public:
   int i;
};

int main() {
   Test* t = new Test();
   cout << t->i;
   delete t;
}
```