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;
```

→ dereference

32

24 and 42

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.

int

0...

this will try to add the address of the int type together → return error

3) If "cout << p3 << endl;" outputs 0x596834,
and "cout << *p4 << endl;" outputs 1, — = 4 bytes
what will be the output of
"cout << p3 + *p4 << endl; cout << p3 - *p4 << endl;"? Explain.
p3 and p4 are of int* type.

① 0x596838 ② 0x596830

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;
}
```

Not reliably because foo: a can also change the value of a.

Example use:

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