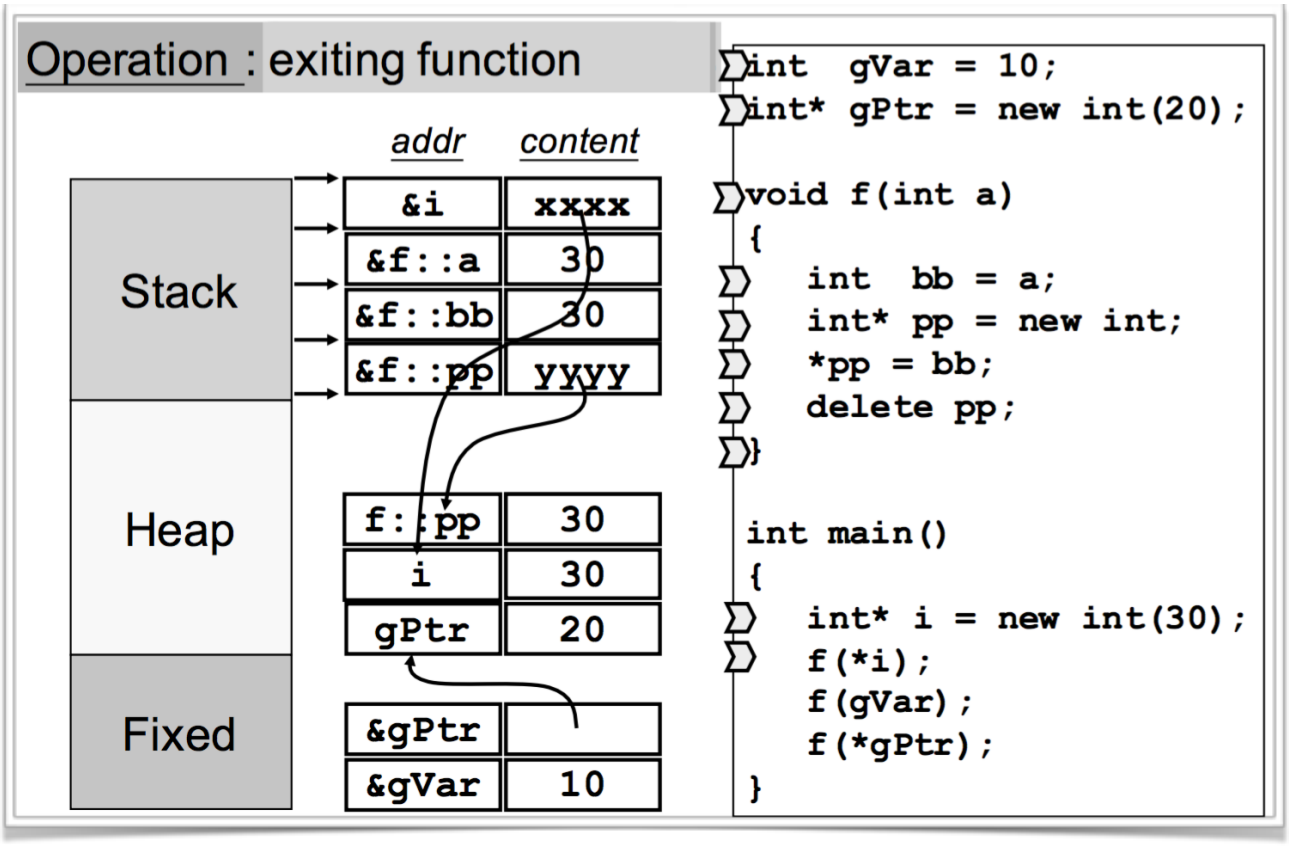

Practice #1 (for lecture note #1)

C++ Review Part I:

The Basic: Variables, Classes, IO Streams

Wednesday, 2019-09-11

Notes and Advices:

1. Submission of the practice is NOT mandatory. The main purpose of these problems is for you to practice and get refreshed on the contents of the lecture note for the coming class meeting. Practice of the problems before class is HIGHLY recommended.
2. However, in case you need to "bargain" your term grade while you are below the next/higher letter grade for a small enough margin, you can talk to me if you have been consistently submitting your practices in time.
3. Note that your submission of the practice will NOT be graded or checked (by me or TA). Again, this is for your practice only.
4. The submissions on NTU CEIBA have a strict deadline. Usually it is before midnight of the class meeting.
5. Practice problems will be, hopefully, uploaded to CEIBA no later than the previous weekend of the class.

Problems // "(pn)" refers to the page #n in the lecture note

1. (p5) Define three variables, say "int a, b, c;"
   - Try to print out their memory addresses.
   - Redefine them to "int b, a, c;" and/or change the order of the printings above. Try to see their memory addresses again.
   - Rename them to "int x, y, z;". Try again.

2. (p8) For the following code:

```
int a, b, c;
while (cin >> a >> b) {
   c = a + b;
   // DO something here…
}
```

   - Try to print out a,b,c's memory addresses.
   - Print out the "content" of these addresses
   - Operate on these variables. Print out their memory addresses and contents again.
   - Try to manipulate on these memory locations. Print out their memory addresses of these variables and the corresponding memory contents again.

3. (p10) Define:

```
int  a;
int *p;
```

   - Print out a and p's memory addresses.

   Following above, do this:

```
p = &a;
```

   - Print out a and p's memory addresses again.
   - Print out the contents of these memory addresses.

4. (p11) Define:

```
int  a = 10;
int &r = a;
```

- Print out a and r's memory addresses.

Following above, do this:

```
int b = 20;
r = b;
```

- Print out a, b and p's memory addresses.
- Print out the contents of these memory addresses.
- What are the values of a, b, and r?

5. (p14) Design a class C of size = 24 Bytes // (hint: use "sizeof()" operator)
   Define objects x, y, z of class C
   - Print out x, y and z's memory addresses.

   Following above, do this:

```
int *a1 = new int;
int *a2 = new int;
C   *c1 = new C;
C   *c2 = new C;
C   *c3 = new C;
```

   - Print out a1, a2, c1, c2 and c3's memory addresses.
   - Print out their memory contents. How do they look different from x, y, and z?

6. (p16) Repeat Practice #1, except that defining "int a, b, c;" in a function f().
   Call f() in main().
   - Try to print out their memory addresses.
   - Call f() multiple times. Check their memory addresses.

   Following above, also do this in f():

```
int *p = new int;
int *q = new int;
```

   - Observe the differences from the above.

7. (p32) Define a class C as follows:

```
class C {
public:
   void printAddr() const { … }
private:
   int  _a;
   int  _b;
};
```

   - Inside member function printAddr(), print out the memory addresses of _a and _b.
   - In main(), define two class C objects c1, c2. Print out the memory addresses of c1 and c2.

8. (p33) For the class C in Practice #7:
   - Define a constructor to initialize _a and _b.
   - Define a member function printData() to print out the contents of _a and _b.
   - In main(), define two class C objects c1, c2. Pass in values to their constructors properly to initialize their data members.
   - Use c1 and c2 to call printData() to see if they are properly initialized.
   - Use c1 and c2 to call printAddr(). Observe their memory addresses.

9. (p34) Define class A and class B as follows:

```
class B {
public:
   B(…) {…}  // define B's constructor
private:
   int  _bd;
};
class A {
public:
   A(…) {…}  // define A's constructor
private:
   int  _ad;
   B    _b;
};
```

   - Define no other member function for class A and class B. Can you properly initialize "B _b" in A?.

10. (p41) Define class A and class B as follows:

```
class C; // Just declared
class B {
   int  _bd;
};
class A {
   B    _b;
   C   *_c;
};
```

   - In main(), declare an object "A a;". Why can this be compiled successfully without defining class C?
   - Observe their memory addresses.
   - What's the content of "a._c"? What does it imply? Who assigns and/or initializes it?

11. (p49) For the examples in page 49, observe the contents and memory addresses of all the variables, and the contents of the memory addresses the variables point to, if applicable.
   - What are the addresses of arr[0] and arr[9]? Which one is larger? Why?
   - Observe the differences of address spaces between arr[i], p2, and p2[i].
   - Observe the memory allocation of two dimensional array.