Ocean Lu
CS 4080.02
Professor Yang
10/15/2019

Assignment #4

1. C++ Given three pieces of C++ code, run and observe output, then answer the questions below. Note: in order to run the code you may need to edit the code such as including libraries, adding main functions etc. Also, if any syntax error spotted in the code please let instructor know or make correct based on your best judgement.

(1)
```
void swap (int x, int y) {
        int t = x;
          x = y;
          y = t;
          return;
}
int a = 10, b = 20;
swap(a,b);
//now print a and b
// a = 10, b = 20
```

(2)
```
void swap (int *px, int *py) {
          int t = *px;
          *px = *py;
          *py = t;
          return;
          }
int  a = 10, b = 20;
swap (&a, &b);
//print a and b
// a = 20, b = 10
```

(3)
```
void swap (int& x, int& y) {
          int t = x;
          x = y;
          y = t;
          return;
          }
int a = 10, b = 20;
swap(a,b);
//now print a and b
// a = 20, b = 10
```

Question (a): what is the parameter passing method for each of the above code?
(1) has a parameter passing method utilizes values, taking in integers.
(2) has a parameter passing method utilizes memory location, taking in pointers.
(3) has a parameter passing method utilizes memory location, taking in references.

Question (b): Run the code and observe the results. Which code(s) swap two values? Which code(s) don't? Why – briefly explain.
The (2) and (3) swap values. (1) does not swap the values. This is because in (1), the values a and b are passed to swapped, and the function does swap them, but nothing has changed in the main function. In (2), the pointers point at the integers and when swap is called, it calls the pointers which mean the integers. In (3), they use references with the integers.

2. Python

```
def swap (x, y):
    t = x
    x = y
    y = t
a =10
b =20
swap(a,b)
print(a,b)
# a = 10, b = 20
```

(a) What parameter passing method used in the above example?
The parameter passing method utilizes values.

(b) Run the above program. Based on the output please state whether the function swaps two values or not.
The program does not swap the two values.

(c) If the above function doesn't swap two values, write one line of Python code that would be able to swap two values.

```
def swap (x, y):
    t = x
    x = y
    y = t
    return x,y
a =10
b =20
a, b = swap(a,b)
print(a,b)
```

one line of python code that would be able to swap the values without function calling would be:

```
a,b = b,a
```

3. Java: Write a Java function to swap two integers.

```
public static void swap(IntHolder a, IntHolder b) {
      int temp = a.value;
      a.value = b.value;
      b.value = temp;
   }
   public static void main(String []args){
      IntHolder a = new IntHolder(10);
      IntHolder b = new IntHolder(20);
      swap(a, b);
      System.out.println(a.value + " " + b.value);
    }
```

4. Examine the C++, Python, and Java codes that do swap two values, which one you think is the best, which one you think is the worse – Note: identify the language evaluation criteria you used, but no need to justify.

   I do believe that the Python is the best language to implement a swap function. It has a very simplistic characteristic, that has very good readability, writability, and reliability.

5. Looking at the following Python code that attempt to swap L[i]'s value with L[j]'s value.

```
def swap (Lst, indexA, indexB) :
   t = Lst[indexA]
   Lst[indexA] = Lst[indexB]
   Lst[indexB] = t

L = [1,2,3,4,5,6,7,8,9,10]
i=3
j=7
print(L[i], L[j])
swap(L,i,j)
print(L[i],L[j])
```

   Question (a): what parameter passing methods are used in each method? (note: multiple parameter passing methods may be used, in this case indicate the parameter name and its corresponding parameter passing method respectively.)

   The three parameters taken in are not defined with type checking parameters. In our code, we pass the array, as well as the integers i and j, which will be used as indexes for L.

   Question (b): will this code swap L[i]'s value with L[j]'s value? Run the code and observe the result.

   The result does swap L[i]'s and L[j]'s value, which prints out the original array first, and then prints out the results later—which looks like:

   4 8
   8 4

6. Write a C++ or Java code that does similar swap action as that in Problem 5.

```
public static final <T> void swap(T[] a, int i, int j) {
    T t = a[i];
    a[i] = a[j];
    a[j] = t;
}

public static void main(String[] args) {
      int[] L = new int[] {1,2,3,4,5,6,7,8,9,10};
      int i = 3;
      int j = 7;
      System.out.println(L[i] + " " + L[j]);
      swap(L, i, j);
      System.out.println(L[i] + " " + L[j]);
  }
```