Drew Pulliam – DTP180003

Assignment 6

CS 4337.0U2

1. Part A

j =-3;

for (int i=0;i<3;i++){

if((j+2)==3 || (j+2)==2){

j--;

}else if((j+2)==0){

j+=2;

}else{

j=0;

}

if(j<=0)

j=3-i;

}

Part B

switch(k)

{

case 1:

case 2:

j = 2 \* k - 1;

break;

case 3:

case 5:

j = 3 \* k + 1;

break;

case 4:

j = 4 \* k - 1;

break;

case 6:

case 7:

case 8:

j = k - 2;

break;

}

1. See folder “q2” for source code.
2. See folder “q3” for source code.
3. C++ namespaces do not have a hierarchy system, but java packages do have a hierarchy system. Both packages and namespaces are ways to import classes, in other words they are containers for classes. Both are used to make code more readable by abstracting away some code. Without packages and namespaces, it becomes much harder to import new classes, which would incentivize much larger individual source files.
4. A. Python local variables are all dynamically allocated, this can be seen best by the fact that variables do not need to declare a type, and they can change type as well.

X = 5

X = “hello”

These two lines are legal in python, and after both of them, the variable X is set to the string “hello”.

B. Python functions (subprograms) can absolutely define other functions inside of themselves.

def function1(): # outer function

print ("Hello from outer function")

def function2(): # inner function

print ("Hello from inner function")

function2()

function1()

Function2 is defined and called inside of function1

C. Python automatically figures out parameter passing method for you. What this means is when an object is passed it acts as a pass by reference, but when an immutable argument is passed it will simply be pass by value (because the argument cannot be modified anyways).

D. Python functions do not check variable types because python does not really have strict variable types. In part A I showed an example of how a variable X can change type extremely easily.

E.

F. Python functions cannot be overloaded, simply because python does not specify types in the function definition. Technically this means that functions can be called with different types, meaning the affect of having an overloaded function is achieved.

G. Technically all python functions are generic. This is because python does not specify variable types in the function definition. This means that functions can be called with any variable type (although some might cause errors within the function).

H. Python does allow nested functions as discussed previously, and it also supports closures with these nested functions. This means that the closure will remember values in scopes that are not necessarily present in memory at the moment.