Worksheet Assessment 2

Student Name: Noreen Lenihan
Student ID: 13204807
Course: COMP47160 Advanced Java & DSA
Course Instructor: Dr. Rem Collier
Due date: Fri, 7 Feb

Question 2

<algorithm cost>
= <cost of assignment>+ <cost of loop> + <cost of return>
= 1 + <cost of loop> + 1

<cost of loop>
= <cost of initial assignment> + <cost of guard> +
(<cost of guard> + <cost of increment> +
<cost of inner statements>) * <number of iterations>
= 1 + 2 + (2 + 2 + 3) * n
= 3 + 7n

```
Algorithm program()
Input: none
Output: none
print("enter a number: ")          // Print (1)
num ← read()                       // Assignment (1), Read (1)
print("you entered: ")             // Print (1)
if (num < 10000) print("0")        // Compare (1), Print (1)
if (num < 1000) print("0")         // Compare (1), Print (1)
if (num < 100) print("0")          // Compare (1), Print (1)
if (num < 10) print("0")           // Compare (1), Print (1)
println(num)                       // Print (1)
```

Total Operation Count: 13
Big-Oh Estimation Running time: O(1)

Question 5

```
Algorithm program()
Input: none
Output: none
```

```
i ← 20

while i >= 0 do                    // Initial assignment (1) + Cost of guard (1) +
                                          (Cost of guard(1) + decrement(2) +
                                   compare(1) + if-print(1) + inner
                                   statements(1))*11  - (no-print-on-first-
                                   iteration(1))
if i < 20 then print(“,”)
print(i)
i ← i – 2
```

Total Operations Count: 67
Big-Oh Estimation Running time: O(1)

Question 8

```
Algorithm program()
Input: none
Output: none

Let A be an array containing {5, 7, 3, 12, 6, 11, 1, 19, 9, 4}  //Assignment(1)
j ← 1                                                          //Assignment(1)
t ← A[0]
                                                  //Assignment + Indexing(2)
while j < 10 do
                                                  (guard(1)
        A[j-1] ← A[j]                             + (guard(1)  + increment(2)
        j ← j + 1
                                                  + inner statement(4))*9
A[j-1] ← t                                        //Assignment, arithmetic, index (3)

for each value, j, in the range 0 to 9 do         //Initial assignment(1) + guard(1) +
                                                  (guard(1) + inner statements(3)+
                                                  increment(2))*10
print(A[j] + “ “)
```

Total operation count: 133
Big-Oh Estimation Running time: O(1)

Question 9

```
Algorithm fn(num, digits)
Input: num and digits
Output: output

output ← “”                          // Assignment(1)
mult = 1                             // Assignment(1)


for each value, j, in the range 1 to digits do                    //Initial assignment(1) + guard(1) + ((inner if-statement)(3) + multi-value-increment(2)  + guard(1) + j-increment(2)) * d

        if num < mult then output ← output + “0”

        mult = mult * 10

output ← output + num                                             // Assignment + concat(2)

return output                                                     // Return (1)

Algorithm program()
Input: None
Output: None

println(fn(75, 6))                                                // Print + method call (2)
```

Total Operations Count: ((8*6) + 9 – (4)) = 53

(We subtracted 4 above, as the if-statement will not evaluate to true on the first two iterations (when num or 75 is NOT less than mult)

Generic operation counting (if values for variables unknown): 8*digits + 9 (or 8n + 9)

Big-Oh Estimation running time: O(n)