WCOM125/ COMP125 Week 1

COMP115/ WCOM115 Revision

April 27, 2017

1. What is the value of `result` when the following code is executed? Show your working using a logic table.

```
int result = 3;
for(int i=1; i <= 20; i+=3) {
        if(i % 4 == 0) {
                result *= 2;
        }
        else {
                result--;
        }
}
```

SOLUTION:

i	i $\leq$ 20	i % 4	i % 4 == 0	result
1	true	1	false	2
4	true	0	true	4
7	true	3	false	3
10	true	2	false	2
13	true	1	false	1
16	true	0	true	2
19	true	3	false	1

Hence, result = 1

2. Write a piece of code that adds the first 100 positive integers (1 to 100) and stores the result in a variable `total`. You must use a loop in order to achieve this.

SOLUTION:

```
int total = 0;
for(int i=1; i <= 100; i++) {
   total = total + i;
}
```

3. What is the value of `result` when the following code is executed? Show your working using a logic table.

```
int total = 0;
for(int i=1; i <= 10; i+=3) {
   for(int k=1; k <= i; k+=3) {
      result++;
   }
}
```

SOLUTION:

outer loop executes for i = 1,4,7,10 for i=1, inner loop executes for k=1 for i=4, inner loop executes for k=1,4 for i=7, inner loop executes for k=1,4,7 for i=10, inner loop executes for k=1,4,7,10

result increases 10 times, and becomes 10

4. What is the value of `result` when the following code is executed? Show your working using a memory diagram.

```
boolean foo(int n) {
        if(n > 5 && n < 10) {
                return true;
        }
        else {
                return false;
        }
}

void setup() {
        int a = 12;
        boolean result = foo(a);
}
```

SOLUTION:

result = false

the memory diagram should show two memory scopes (one for setup() and the other for foo(a). for the one in foo(a), the value of `a` should be copied into `n`)

5. Define a function that when passed an integer, returns `true` if it's even (divisible by 2) and `false` otherwise.

SOLUTION:

```
boolean isEven(int a) {
        if(a%2 == 0) {
                return true;
        }
        else {
                return false;
        }
}
```

6. What is the value of `result` when the following code is executed?

```
int bar(int a, int b) {
        if(a > b)
                return a;
        else
                return b;
}

void setup() {
        int result = bar(bar(4,2), bar(3,6));
}
```

SOLUTION:

result = bar(4, 6) = 6

7. Define a function that when passed an integer (call it `num` in the scope of the function call), returns the sum of the first `num` positive integers. You may assume `num` > 0. For example, if `num = 4`, function should return 10 (1+2+3+4 = 10).

SOLUTION:

```
int sum(int num) {
   int result = 0;
   for(int i=1; i <= num; i++) {
      result = result + i;
   }
   return result;
}
```

8. What changes must you make to the function `sum` defined above if the assumption (`num` > 0) is no longer valid. What value do you think should be returned for `num` ≤ 0

SOLUTION:

```
int sum(int num) {
   if(num <= 0) {
      return 0;
   }

   int result = 0;
   for(int i=1; i <= num; i++) {
      result = result + i;
   }
   return result;
}
```

9. Define a function that when passed two integers (call them `x, n` in the scope of the function call), returns the x^n (`x * x * x ... n times`). You may assume `n` > 0. For example, if `x = 2, n = 4`, function should return 16 ($2^4 = 2 * 2 * 2 * 2 = 16$).

SOLUTION:

```
int power(int x, int n) {
   int result = 1;
   for(int i=1; i <= n; i++) {
      result = result * x;
   }
   return result;
}
```

10. Create an array that holds 500 integers. Using a loop, populate the array, such that,

    - the first item is 5
    - the second item is 7
    - the third item is 9
    - the fourth item is 11
    - and so on

```
int[] a = new int[500];
int val = 5;
for(int i=0; i < a.length; i++) {
        a[i] = val;
        val = val + 2;
}
```

11. Create an array that holds 100 real numbers. Using a loop, populate the array, such that,

    - the first item is 7.5
    - the second item is 7.45
    - the third item is 7.40
    - the fourth item is 7.35
    - and so on

```
double[] a = new double[500];
double val = 7.5;
for(int i=0; i < a.length; i++) {
        a[i] = val;
        val = val - 0.05;
}
```

12. Consider the following function definition,

```
1 float square(float n) {
2         return n*n;
3 }
```

Write one or two statements that sit inside the `setup()` function that calls the function `square` to compute 5^2, and stores the returned value in a variable `result`. You must declare the variable `result` to an appropriate data type.

SOLUTION:

```
1 float result = square(5);
```

13. Define a function `total` that when passed an integer array, returns the sum of all the items in the array. Return 0 if the array is `null`

SOLUTION:

```
int total(int[] a) {
        if(a == null)
                return 0;
        int result = 0;
        for(int i=0; i < a.length; i++) {
                result = result + a[i];
        }
        return result;
}
```

14. What changes must you make to the function `total` defined above if you want to add **only the positive items**?

SOLUTION:

```
int totalEven(int[] a) {
        if(a == null)
                return 0;
        int result = 0;
        for(int i=0; i < a.length; i++) {
                if(a[i] > 0) {
                        result = result + a[i];
                }
        }
        return result;
}
```

15. What changes must you make to the function `total` defined above if you want to add **only the items in a specific range**. Say items that lie between:

    - 1 and 6, or,
    - 50 and 100

SOLUTION:

```
int totalInRange(int[] a, int low, int high) {
        if(a == null)
                return 0;
        int result = 0;
        for(int i=0; i < a.length; i++) {
                if(a[i] >= low && a[i] <= high) {
                        result = result + a[i];
                }
        }
        return result;
}
```

16. Define a function `highest` that when passed an integer array, returns the highest value in the array. Return 0 if the array is`null` or if the array is empty.

SOLUTION:

```
int highest(int[] a) {
        if(a == null)
                return 0;
        if(a.length == 0) //empty array
                return 0;

        int result = a[0]; //assume first item is the highest
        for(int i=1; i < a.length; i++) { //start from second
           item
                if(a[i] > result) {
                        result = a[i];
                }
        }
        return result;
}
```

17. What changes must you make to the function `highest` defined above if you want to return the **smallest item**.

SOLUTION:

```
int smallest(int[] a) {
        if(a == null)
                return 0;
        if(a.length == 0) //empty array
                return 0;

        int result = a[0]; //assume first item is the highest
        for(int i=1; i < a.length; i++) { //start from second
           item
                if(a[i] < result) {
                        result = a[i];
                }
        }
        return result;
}
```

18. Define a function `highestIndex` that when passed an integer array, returns the **index of** the highest value in the array. Return -1 if the array is`null` or if the array is empty.

SOLUTION:

```
int highestIndex(int[] a) {
        if(a == null)
                return 0;
        if(a.length == 0) //empty array
                return 0;

        int result = 0; //assume first item is the highest
        for(int i=1; i < a.length; i++) { //start from second
           item
                if(a[i] > a[result]) {
                        result = i;
                }
        }
        return result;
}
```

19. Which function is more powerful - `highest`, or `highestIndex`? `highestIndex` as we can get the item directly from index but not the index directly from the item.

20. What changes must you make to the function `highestIndex` defined above if you want to return the **index of the smallest item**.

SOLUTION:

```
int smallestIndex(int[] a) {
        if(a == null)
                return 0;
        if(a.length == 0) //empty array
                return 0;

        int result = 0; //assume first item is the smallest
        for(int i=1; i < a.length; i++) { //start from second
           item
                if(a[i] < a[result]) {
                        result = i;
                }
        }
        return result;
}
```

21. What changes must you make to the function `highest` defined above if you want to return the highest value **starting from a specific index**. For example, if `a = {40, 80, 30, 50, 70, 20}`, and the index starting at which we should look is 3 (note that a[3] is 50), the function returns 70

SOLUTION:

```
int highest(int[] a, int start) {
        if(a == null)
                return 0;
        if(start < 0 || start >= a.length) //invalid index
                return 0;

        int result = a[start]; //assume first item is the
           highest
        for(int i=start+1; i < a.length; i++) { //start from
           second item
                if(a[i] > result) {
                        result = a[i];
                }
        }
        return result;
}
```

22. What changes must you make to the function `highest` defined above if you want to return the **index of** the highest value **starting from a specific index**. For example, if `a = {40, 80, 30, 50, 70, 20}`, and the index starting at which we should look is 3 (note that a[3] is 50), the function returns 4 (item at index 4 is the highest starting at index 3).

SOLUTION:

```
int highestIndex(int[] a, int start) {
        if(a == null)
                return -1;
        if(start < 0 || start >= a.length) //invalid index
                return -1;

        int result = start; //assume first item is the highest
        for(int i=start+1; i < a.length; i++) { //start from
           second item
                if(a[i] > a[result]) {
                        result = i;
                }
        }
        return result;
}
```

23. Define a function `identical` that when passed two integer arrays, returns `true` if they are identical to each other, `false` otherwise (or if either of the arrays is `null`).

SOLUTION:

```
boolean identical(int[] a, int[] b) {
        if(a == null || b == null)
                return false;
        if(a.length != b.length)
                return false;
        for(int i=0; i < a.length; i++) {
                if(a[i] != b[i]) {
                        return false;
                }
        }
        return true;
}
```

24. **(advanced)** Define a function `withoutFirstDigit` that when passed an integer n, returns the number without the first digit. You may assume that m is more than 0. For example, if $m = 7129$, function returns 129.

SOLUTION:

```
int withoutFirstDigit(int m) {
  int result=  0;
  int power = 1;
  while (m != 0) {
    if (m > 9) {
      result = m%10 * power + result;
    }
    power*=10;
    m/=10;
  }
  return result;
}
```

OR

```
int withoutFirstDigit(int m) {
        if(m == 0)
                return 0;
        m = Math.abs(m);
        String s = m+"";
        int result = Integer.parseInt(s.substring(1));
        return result;
}
```

25. Draw the memory diagram that captures the transactions when the following code executes.

```
1 int[] a = {1, 7, 2};
2 int[] b = a;
3 int[] c = new int[a.length];
4 for(int i=0; i < a.length; i++) {
5         c[i] = a[i];
6 }
```

SOLUTION: