CSC 172 HW0 **Dr. Chays**

Name: ____

Answer ALL questions. Show ALL work. CLEARLY indicate your final answer.

1. What output is produced by the following code fragment?

```
final int MAX = 4;
int prod = 1, num = 1;

while (num <= MAX)
{
   prod = prod * num;
   num ++;
}

System.out.println (prod);
System.out.println (num);
```

2. Transform the highlighted code from the previous question into an equivalent for loop.

3. Write a method named *isValid* which receives one parameter of type int representing a user choice, and returns true if it is 1, 2 or 3; false otherwise.

4. Write a do loop that reads in a user choice; if it is invalid, print an error message and read the input again. (For full credit, call/invoke the method in the previous question appropriately.)

5. Given integer variables hours and minutes (already declared and defined), write Java code to print:

The time is value of hours : value of minutes

Example: If hours is 10 and minutes is 45, the output should be: **The time is 10:45** (For full credit, the output should be appropriate even if minutes is a single digit, e.g. if hours is 7 and minutes is 2, the output should be: **The time is 7:02**)

6. Show the output, including proper spacing.

```
for (int j = 1; j <= 3; j++)
{
        for (int k = 1; k <= 3; k++)
         {
              System.out.print (j + k);
         }

        System.out.println();
}
```

7. Write a method to compute and return the value of the *average* element of an int array. The method is passed the int array and an int value that denotes the number of elements in the array. For example, if the array has 4 elements: {60, 70, 80, 90}, the method should return (60 + 70 + 80 + 90) / 4 = 75. If the array has no elements, return -1.

8. Define a class named Item that includes:
- attributes representing an item’s description and unit price (note: the unit price is the cost of one item and may have a fractional part, e.g. 4.50).
- a constructor that is passed 2 parameters and initializes the corresponding instance variables.
- a method named computeTotalCost that receives one parameter representing the number of items purchased, and computes and returns the total cost of the items purchased, including sales tax which is 8.25%.
- a method named equals that returns true if this item has the same description and unit price as that of the Item object received as the parameter; false otherwise.
- a method named toString that receives no parameters and returns a formatted String representation of this Item that includes the values of all attributes.
- **driver code** (a static test method of the same class, or a main method in another class) that instantiates (creates) 2 objects of type Item and invokes the above methods appropriately. For each method that you call (i.e. the computeTotalCost and equals methods), print the actual result returned by the method and the expected result.
Print only in your driver code!

9. **What is the exact output produced by running the method test?**

```
public class TestSwap
{
    public void swap (int x, int y)
    {
        int temp;
        temp = x;
        x = y;
        y = temp;
        System.out.println("Inside swap version 1:");
        System.out.println("x = " + x);
        System.out.println("y = " + y);
    }

    public void swap (int[] a, int i, int j)
    {
        int temp;
        temp = a[i];
        a[i] = a[j];
        a[j] = temp;
        System.out.println("Inside swap version 2:");
        System.out.println("a[" + i + "] = " + a[i]);
        System.out.println("a[" + j + "] = "+ a[j]);
    }

    public void printArray (int[] a)
    {
        System.out.print("Array elements: ");

        for (int i = 0; i < a.length; i++)
            System.out.print(a[i] + " ");

        System.out.println();
    }

    public static void test()
    {
        TestSwap t = new TestSwap();
        final int ARRAY_SIZE = 3;
        int[] arr = new int[ARRAY_SIZE];
        arr[0] = 4;
        arr[1] = 7;
        arr[2] = 8;

        int x = 0, y = 2;

        t.printArray (arr);
        t.swap (x, y);
        System.out.println ("Inside test:");
        System.out.println ("x = " + x);
        System.out.println ("y = " + y);
        t.printArray (arr);
        t.swap (arr[x], arr[y]);
        t.printArray (arr);
        t.swap (arr, x, y);
        t.printArray (arr);
    }
}
```

10. Reversing the elements of an array involves exchanging the corresponding elements of the array: the first with the last, the second with the next to the last, and so on, all the way to the middle of the array. For example, if the array has 4 elements: {80, 85, 95, 100}, we want to reverse the elements so that the array becomes {100, 95, 85, 80}.

Given an array a, write a loop that reverses the elements of the array. You may declare additional variables as needed. (For full credit, invoke one of the swap methods from question 9 appropriately.)

11. Write a method that receives a String as a parameter, creates a new String that contains the same characters in the specified String but in reverse order, and returns that String. For example, if the String received as a parameter is "loop", the method should return "pool". Your method should work for any String.

You will need to appropriately invoke the charAt method of the String class. Its description in the Java standard library is:

`char` **charAt**`(int index)`

> Returns the `char` value at the specified index.