CMP-326 Total 120 Points Test 2

1. (10 Points) What is the output of the following code:

```
public static void main(String[] args) {
        int s = 1;
        int t = 1;
        for ( int i = 1 ; i < 21 ; i += 2 ) {
                s = s + i;
                for ( int j = i ; j > 0 ; j -= 3 ) {
                        t = t + (i - j);
                }
                s = s + t;
                System.out.println("t = " + t);
        }
        System.out.println("s = " + s);
}
```

```
t = 1
t = 1
t = 4
t = 13
t = 22
t = 40
t = 70
t = 100
t = 145
t = 208
s = 705
```

2. (20 Points) Given the following classes, show the output from running the Test class and give an explanation:

```
public class Vehicle {
   private String name =
                     new String("No Vehicle Name");
    public Vehicle() {
      System.out.println("New Vehicle: " + name);
   }
    public Vehicle(String name) {
      this.name = new String(name);
       System.out.println("New Vehicle: " + name);
   }
    public void drive() {
       System.out.println("Vehicle drive: " + name);
   }
}
```

```
public class Car extends Vehicle {
    private String name =
                         new String("No Car Name");
    public Car() {
      super("No Car Name");
       System.out.println("New Car: " + name);
   }
    public Car(String name) {
      super();
       this.name = new String(name);
       System.out.println("New Car: " + name);
   }
    public void drive() {
       System.out.println("Car drive: " + name);
   }
}
```

```
public class SportsCar extends Car {
    private String name =
                         new String("No Sports Car Name");
   public SportsCar() {
      super("No Sports Car Name");
      System.out.println("New SportsCar: " + name);
   }
    public SportsCar(String name) {
      super();
      this.name = new String(name);
      System.out.println("New Vehicle: " + name);
   }
    public void drive() {
      System.out.println("Sports Car drive: " + name);
   }
}
```

```
public class Question2 {
        public static void main(String args[]) {
                Vehicle v, v1, v2;
                Car c, c1, c2;
                SportsCar sc, sc1, sc2;
                v1 = new Vehicle("Veronica");
                v2 = new Vehicle();
                c1 = new Car("Carlos");
                c2 = new Car();
                sc1 = new SportsCar("Sport");
                sc2 = new SportsCar();
                v1.drive();
                v2.drive();
                c1.drive();
                c2.drive();
                sc1.drive();
                sc2.drive();
                v = c1;
                v.drive();
                c = sc1;
                c.drive();
        }
}
```

```
New Vehicle: Veronica
New Vehicle: No Vehicle Name
New Vehicle: No Vehicle Name
New Car: Carlos
New Vehicle: No Car Name
New Car: No Car Name
New Vehicle: No Car Name
New Car: No Car Name
New Sports Car: Sport
New Vehicle: No Vehicle Name
New Car: No Sports Car Name
New Sports Car: No Sports Car Name
Vehicle drive: Veronica
Vehicle drive: No Vehicle Name
Car drive: Carlos
Car drive: No Car Name
Sports Car drive: Sport
Sports Car drive: No Sports Car Name
Car drive: Carlos
Sports Car drive: Sport
```

3. (30 Points) Given an array of String. Each element in the array contains a String object. Example:

“Sameh”	“Eliot”	“Sarah”	“Dalia”	“Nidal”

Write a method with the following signature:

```
public static String combine(String[] strings)
```

Which combines the Strings to form one String that would contain the last letters of each String, starting with the last name, followed by the second to last letter from each string again starting with the last name, etc.... You may assume that all Strings have the same length, but you may not assume that the array only has 5 entries. The above array would return the following String:

“lahthaiaoedlrimiaalaNDSES”

```
public class Question3 {

        public static String combine(String[] strings) {
                // define the return variable
                String s = new String();

                // find the length of all the Strings by looking at the
                // length of the first String
                int strLength = strings[0].length();

                // i represents the character position within each word
                for ( int i = strLength-1 ; i >= 0 ; i--) {
                        // j represents which word within the array
                        for ( int j = strings.length-1 ; j >= 0 ; j-- ) {
                                // get the i'th character from the j'th word
                                s = s.concat(strings[j].substring(i,i+1));
                        }
                }
                // return the combined String
                return (s);
        }

        public static void main(String[] args) {
                String[] strings = new String[5];

                strings[0] = new String("Sameh");
                strings[1] = new String("Eliot");
                strings[2] = new String("Sarah");
                strings[3] = new String("Dalia");
                strings[4] = new String("Nidal");

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

4. (10 Points) Write a `for` loop to compute the sum $1^1 + 2^2 + 3^3 + 4^4 + 5^5 + ... + n^n$. Assume that `n` is a variable that has already been defined.

```
long n = 10;
long sum = 0;

for ( int i = 1 ; i <= n ; i++ ) {
      sum += Math.pow(i,i);
      System.out.println("for " + i + " the sum = " + sum);
}
```

5. (10 Points) Show the output from running the Test class:

```
public class Question5 {
	public static void main(String args[]) {
		String s1 = new String("josue");
		String s2 = new String("joseph");
		String s3 = new String("Jose");
		String s4 = s2;

		if (s1.substring(0,3).equals(s3.substring(0,3))) {
			System.out.println("Test1 Is A Success");
		} else {
			System.out.println("Test1 Is A Failure");
		}

		if (s2.substring(1,4).equals(s3.substring(1,4))) {
			System.out.println("Test2 Is A Success");
		} else {
			System.out.println("Test2 Is A Failure");
		}

		if (s2 == s4) {
			System.out.println("Test3 Is A Success");
		} else {
			System.out.println("Test3 Is A Failure");
		}

		if (s3.toLowerCase().substring(0,3).equals(s1.substring(0,3))) {
			System.out.println("Test4 Is A Success");
		} else {
			System.out.println("Test4 Is A Failure");
		}
	}
}
```

```
Test1 Is A Failure
Test2 Is A Success
Test3 Is A Success
Test4 Is A Success
```

6. (40 Points) Write a complete Java class named `MyClass` that has the following `private` attributes:

a. myInts, an array of `int`‘s that has a maximum capacity of 100.
b. numInts, an `int` variable that keeps track of the number of elements in myInts.

And the following methods:

a. `public` MyClass() – Constructor that intitializes myInts and numInts.
b. `public int` addInt(int i) – Adds i to myInts and updates numInts. Returns the index where i was added. If there is no room in the array, expand the array by adding another 100 spaces.
c. `public int` findFirst(int i) – Find the first occurrence of i in myInts and return its index, return -1 if not found.
d. `public int` findLast(int i) – Find the last occurence of i in myInts and return its index, return -1 if not found.
e. `public int` getInt(int i) – Return the integer at index i if it exists, return -9999 otherwise.
f. `public boolean` isFull() – Returns true if myInts is full, false otherwise.
g. `public boolean` isEmpty() – Returns true if myInts is empty, false otherwise.

```
public class Question6 {

      private final int MAX_INTS = 100;

      private int[] myInts;
      private int numInts;

      // Constructor: initializes myInts and numInts
      public Question6() {
            myInts = new int[MAX_INTS];
            numInts = 0;
      }

      // Adds 'i' to numInts. If the array isn't big enough,
      // then this method expands the array
      public int addInt(int i) {
            int index = -1;
            if (isFull()) {
                  expandMyInts();
            }
            index = numInts;
            numInts++;
            myInts[index] = i;
            return index;
      }
```

```
// find the first occurrence of 'i' in numInts and return its index
// if the number does not exist in the array , return -1
public int findFirst(int i) {
	int index = -1;
	if (!isEmpty()) {
		for (int j = 0 ; j < numInts ; j++ ) {
			if (myInts[j] == i) {
				index = j;
				break;
			}
		}
	}
	return index;
}

// find the last occurrence of 'i' in numInts and return its index
// if the number does not exist in the array, return -1
public int findLast(int i) {
	int index = -1;
	if (!isEmpty()) {
		for (int j = (numInts-1) ; j >= 0 ; j-- ) {
			if (myInts[j] == i) {
				index = j;
				break;
			}
		}
	}
	return index;
}

// if 'i' is a valid index, then return the integer at that index
// otherwise, return -9999
public int getInt(int i) {
	int value = -9999;
	if (i < numInts) {
		value = myInts[i];
	}
	return value;
}

// returns 'true' if numInts is empty, 'false' otherwise
public boolean isEmpty() {
	boolean answer = false;
	if (numInts == 0) {
		answer = true;
	}
	return answer;
}

// returns 'true' if numInts is full, 'false' otherwise
public boolean isFull() {
	boolean answer = true;
	if (numInts < myInts.length) {
		answer = false;
	}
	return answer;
}
```

```
    // expands numInts by an additional 'MAX_INTS' entries
    // copies all entries from the old array to the new array
    private void expandMyInts() {
        int newLength = myInts.length + MAX_INTS;
        int[] newMyInts = new int[newLength];

        for (int i = 0 ; i < myInts.length ; i++ ) {
            newMyInts[i] = myInts[i];
        }

        myInts = newMyInts;
        System.out.println("Expanded myInts to length " + myInts.length);
    }

    // test code for verifying the methods of Question6
    public static void main(String[] args) {
        Question6 q6 = new Question6();

        for ( int i = 0 ; i < 6 ; i++ ) {
            for ( int j = 0 ; j < 100 ; j++) {
                System.out.println(j + " - Added at index - " + q6.addInt(j));
            }
        }

        for ( int i = 0 ; i < 100 ; i++ ) {
            System.out.println("findFirst(" + i + ") = " + q6.findFirst(i));
            System.out.println("findLast (" + i + ") = " + q6.findLast(i));
        }

        System.out.println("findFirst(101) = " + q6.findFirst(101));
        System.out.println("findLast (101) = " + q6.findLast(101));

        for ( int i = 0 ; i < 600 ; i++ ) {
            System.out.println("getInt(" + i + ")  = " + q6.getInt(i));
        }
    }

}
```