

**THE UNIVERSITY OF VICTORIA**  
**CSC 115: MIDTERM EXAMINATION**  
**INSTRUCTOR: CELINA BERG**  
**FEBRUARY 6, 2019**

Name (Last, First): \_\_\_\_\_ Student #: \_\_\_\_\_

Signature: \_\_\_\_\_

**Notes about this examination**

1. You have **50 minutes** to write this examination.
2. The exam is out of 45 marks. Each question indicates the number of marks allocated for that question.
3. Write your name, student #, signature, and exam ID using a pen.
4. All code must be written in the Java programming language. Use good programming practices to get full marks.
5. We will not answer questions during the exam. If you feel there is an error or ambiguity in the question, write your assumption on the page and continue to answer the question.
6. The exam is open book, open notes but you may only use the notes or book you bring. That is, you cannot share books or notes with other students.
7. No electronic equipment is allowed including cell phones, calculators, or smart watches.
8. In the interest of time, it is not necessary to include comment statements in your code, unless specifically asked for.

Question	Mark	Max
1		10
2		10
3		10
4		10
5		5
Total		45

**Q1. (10 marks)** What is the output of the following program?

```
public class Question1 {
    public static void main(String[] args) {
        int[] data = {2, 3, 1, 7};
        int count = 4;
        foo( data );

        System.out.println(data[0]);
        System.out.println(data[3]);
        System.out.println(count);
    }

    public static int foo(int[] array) {
        int count = 0;

        for (int i = 0; i < array.length; i++) {
            // Remember that array[i] % 2 returns
            // the remainder when dividing array[i] by 2
            // For example:
            // 3 % 2 is 1
            // 4 % 2 is 0
            if (array[i] % 2 != 0) {
                array[i] *= 2;
                count++;
            }
        }
        return count;
    }
}
```

**Q2. (10 marks)** Implement the following method according to the specification given.

```
// Purpose:
//   Return true if the numbers in the parameter array are in sorted order
//   from smallest to biggest, false otherwise
// Examples:
//   if a is {1,2,3,4,5} then isSorted (a) returns true
//   if a is {1,2,4,5,10} then isSorted (a) returns true
//   if a is {} then isSorted (a) returns true
//   if a is {2,2,3,4} then isSorted (a) returns true
//   if a is {2,12,3,4} then isSorted (a) returns false
//
public class Question2 {
    public static void main (String[] args) {
        int[] a = {1,2,4,5};
        System.out.println(isSorted(a));
    }
    public static boolean isSorted (int[] array) {
        // write your code here

    }
}
```

**Q3. (10 marks)** What is the output of the following program?

**NOTE:** the `ClothingItem` class is on the next page.

```
public class Question3 {  
  
    public static void main (String[] args) {  
  
        ClothingItem c0;  
        System.out.println("C");  
  
        ClothingItem c1 = new ClothingItem(35.99, "233WW");  
        ClothingItem c2 = new ClothingItem(45.45, "341XY");  
        c0 = new ClothingItem();  
  
        System.out.println("D: " + c0.getBarcode());  
        System.out.println("E: " + c1.getBarcode());  
        System.out.println("F: " + c2.getBarcode());  
  
        c2 = c0;  
        System.out.println("G: " + c0.getBarcode());  
        System.out.println("H: " + c1.getBarcode());  
        System.out.println("I: " + c2.getBarcode());  
  
        c0 = c1;  
        System.out.println("J: " + c0.getBarcode());  
        System.out.println("K: " + c1.getBarcode());  
        System.out.println("L: " + c2.getBarcode());  
    }  
}
```

Write your answer in this box:

```
public class ClothingItem {
    double price;        // in dollars
    String barcode;      // unique barcode

    public ClothingItem () {
        price    = 0.0;
        barcode = "unknown";
        System.out.println("A");
    }

    public ClothingItem (double price, String barcode) {
        this.price = price;
        this.barcode = barcode;
        System.out.println("B");
    }

    public void setPrice (double price) {
        this.price = price;
    }

    public double getPrice () {
        return price;
    }

    public void setBarcode (String barcode) {
        this.barcode = barcode;
    }

    public String getBarcode () {
        return barcode;
    }

    public String toString () {
        return barcode + ": $" + price;
    }

    public int compareTo (ClothingItem otherItem) {
        // you will implement this method in Q4
    }
}
```

**Q4. (10 marks)** Complete the implementation of the following method to be include in the `ClothingItem` class defined in Question 3 according to the following documentation:

```
/*
 * Parameters: (ClothingItem) otherItem
 * Purpose: determines whether the price of otherItem is less than,
 * equal to or greater than the price of this ClothingItem
 * Returns:(int) -1 if this item price is less than otherItem price,
 *              0 if they are equal and
 *              1 if this item price is greater than otherItem price
 */
public int compareTo (ClothingItem otherItem) {
```

**This page intentionally left blank. You can use this page for scratch work. If you write any answers on this page you must indicate this here and on the question itself.**

**Q5. (5 marks)** The following program includes the `ClothingItem` class defined in Question 3 and the `ClothingItemList` class defined on the next page.

HINT: the error message tells you that you can limit your debugging to the `Question5` and `ClothingItem` classes.

This code compiles but when run, the following error is generated:

```
Exception in thread "main" java.lang.NullPointerException
    at ClothingItemList.addClothingItem(ClothingItemList.java:13)
    at Question5.main(Question5.java:9)
```

- Find and explain the bug that is in this code.
- How would you fix the bug? Use file names and lines numbers to identify where you would make the change and explicitly what the change would be.

```
1. public class Question5 {
2.
3.     public static void main (String[] args) {
4.
5.         ClothingItem c1 = new ClothingItem(35.99, "233WW");
6.         ClothingItem c2 = new ClothingItem(45.45, "341XY");
7.         ClothingItemList l = new ClothingItemList();
8.
9.         l.addClothingItem(c1);
10.        l.addClothingItem(c2);
11.
12.        System.out.println(l);
13.    }
14. }
```



```
1.  public class ClothingItemList {
2.
3.      private static final int INITIAL_SZ = 4;
4.      private ClothingItem[] clothes;
5.      private int          count;
6.
7.      public ClothingItemList() {
8.          ClothingItem[] clothes = new ClothingItem[INITIAL_SZ];
9.          count = 0;
10.     }
11.
12.     public void addClothingItem (ClothingItem c) {
13.         if (count >= clothes.length)
14.             expandAndCopy();
15.         clothes[count] = c;
16.         count++;
17.     }
18.
19.     private void expandAndCopy() {
20.         ClothingItem[] newClothes = new ClothingItem[2*clothes.length];
21.
22.         for(int i=0; i<count; i++)
23.             newClothes[i] = clothes[i];
24.
25.         clothes = newClothes;
26.
27.     }
28.
29.     public String toString() {
30.         String s = "";
31.
32.         for(int i=0; i<count; i++)
33.             s += clothes[i] + "\n";
34.
35.         return s;
36.     }
37.
38.     public int getCount() {
39.         return count;
40.     }
41. }
```