CSC 115
Midterm Exam:
Sections: A01 and A02
Monday, October 5th, 2020

Name:____KEY____(please print clearly!)

UVic ID number:____

Signature:____

Exam duration: 50 minutes

Instructor: Anthony Estey

Students must check the number of pages in this examination paper before beginning to write, and report any discrepancy immediately.

- We will not answer questions during the exam. If you feel there is an error or ambiguity, write your assumption and answer the question based on that assumption.
- Answer all questions on this exam paper.
- The exam is closed book. No books or notes are permitted.
  No electronic devices of any type are permitted.
- The marks assigned to each question and to each part of a question are printed within brackets. Partial marks are available.
- There are ten (10) pages in this document, including this cover page.
- Page 10 is left blank for scratch work. If you write an answer on that page, clearly indicate this for the grader under the corresponding question.
- Clearly indicate only one answer to be graded. Questions with more than one answer will be given a **zero grade**.
- It is strongly recommended that you read the entire exam through from beginning to end before beginning to answer the questions.
- Please have your ID card available on the desk.

Part 1: Array Reference (4 marks)

1. Read through the contents of the code shown below:

```
public static int[] copyArray(int[] arr) {
    int[] copy = new int[arr.length];
    for (int i = 0; i < arr.length; i++) {
        copy[i] = arr[i];
    }
    return copy;
}

public static void changeArray(int[] arr, int val) {
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == val) {
            arr[i] = arr[i] - 1;
        }
    }
}

public static void arrayTest() {
    int[] nums1 = {1, 3, 4, 3, 2};
    int[] nums2 = nums1;
    int[] nums3 = copyArray(nums1);

    changeArray(nums1, 3);
    int[] nums4 = copyArray(nums1);

    changeArray(nums2, 2);

    // What are the contents of the arrays now?
}
```

Your should write the contents of the arrays as they are at the end of the arrayTest method, where the comment says "// What are the contents of the arrays now?".

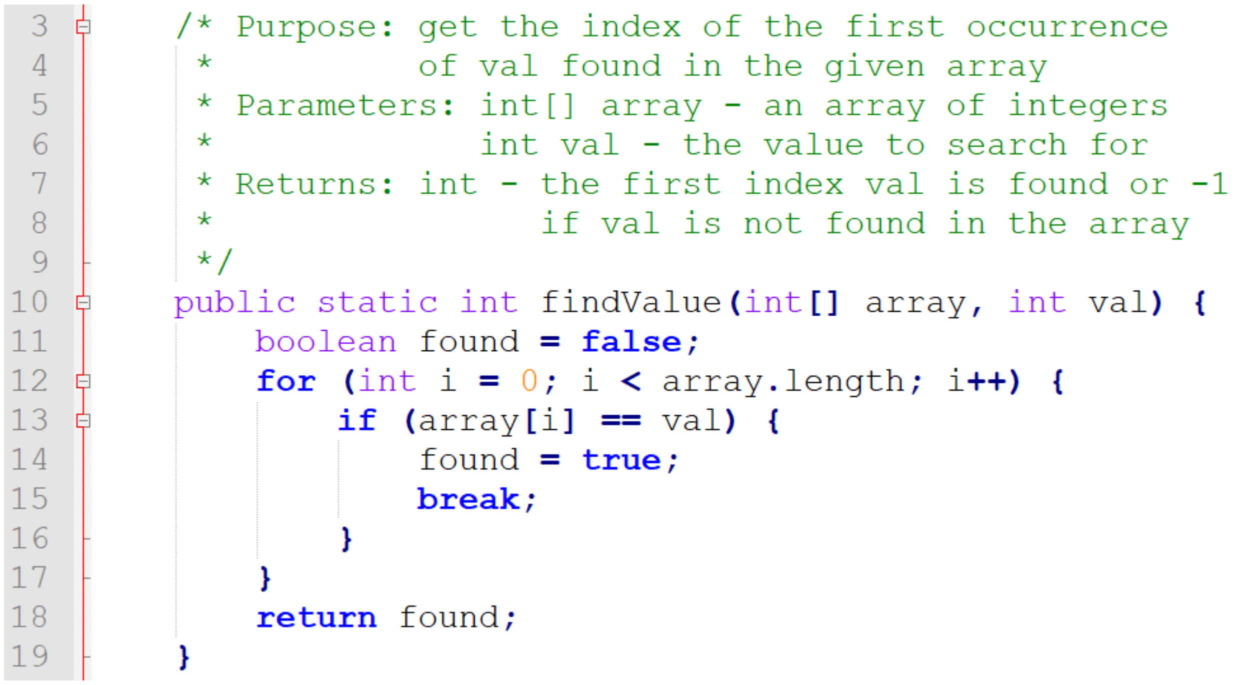
Numbers should be separated by a comma, so for nums1 it initially would be filled out as: [1, 3, 4, 3, 2].

nums1: [1, 1, 4, 1, 1]
nums2: [1, 1, 4, 1, 1]
nums3: [1, 3, 4, 3, 2]
nums4: [1, 2, 4, 2, 2]

Part 2: Debugging (4 marks)

2. The code shown below causes a compile error.

   Describe why, and then explain how to fix it, by providing an example of what code you would write to replace the code given at specific line numbers (shown on the left of the image).



```
 3     /* Purpose: get the index of the first occurrence
 4      *          of val found in the given array
 5      * Parameters: int[] array - an array of integers
 6      *             int val - the value to search for
 7      * Returns: int - the first index val is found or -1
 8      *                if val is not found in the array
 9      */
10     public static int findValue(int[] array, int val) {
11         boolean found = false;
12         for (int i = 0; i < array.length; i++) {
13             if (array[i] == val) {
14                 found = true;
15                 break;
16             }
17         }
18         return found;
19     }
```

Explanation:
The method should return a value of type **int**, but returns a **boolean.**

Fix:
line 11: change **boolean** to **int,** set initial value to **-1**
line 14: change **true** to **i**

Other acceptable fixes:
line 11: declare a new variable (like int position = -1;)
line 14: position = i;
line 18: return position;

2 marks – error explanation
2 marks – fix

Part 3: Testing (6 marks)

3. Given the code shown below, write at least two tests that would go where the line 13 comment is (//Write tests here).

   Briefly state whether the code shown would pass or fail each test you write.

```
 1 public class Question3Tester {
 2     private static int testCount = 0;
 3     private static int testPassCount = 0;
 4 
 5     public static void main(String[] args) {
 6         int[] arr1 = {9, 3, 1, 4, 2};
 7         boolean result = false;
 8         boolean expected = false;
 9 
10         result = contains(arr1, 0);
11         displayResults("arr1 contains 0", result==expected);
12 
13         //Write tests here
14     }
15 
16     /* Purpose: determines whether array contains val
17      * Parameters: int val - the value to search for
18      * Returns: boolean - true if found, false otherwise
19      */
20     public static boolean contains(int[] array, int val) {
21         for (int i=0; i<array.length; i++) {
22             if (array[i] == val) {
23                 return true;
24             } else {
25                 return false;
26             }
27         }
28         return false;
29     }
```

The test provided provides an example of a test that returns false because an integer (0) is not found in the array. Although writing another test to serach for an integer that doesn't exist is good practice, it isn't worth any marks for this exercise, as there are two specific test cases we are looking for:

```
result = contains(arr1, 9);
expected = true;
displayResults("arr1 contains 9", result==expected);
// this will pass as the method will find 9 at index 0 and
// then return true (meaning the value was found)

result = contains(arr1, 3);
expected = true;
displayResults("arr1 contains 3", result==expected);
// this will fail as the method will return false immediately
// after checking if the value at index 0 equals 3. The method
// should continue searching through the array and only return
// false if the value is never found at any index in the array
```

Part marks may also have been awarded for testing an empty array or for creating your own array and performing tests similar to shoe shown above.

3 marks – writing a test that returns true (must search for 9), and statement saying the test will pass. [1 mark for the test, 2 for explanation]

3 marks – writing a test that should return true but does not (3, 1, 4, or 2 passed as a parameter). Must state that the test will fail. [1 mark for the test, 2 for explanation]

Part 4: Interfaces (4 marks)

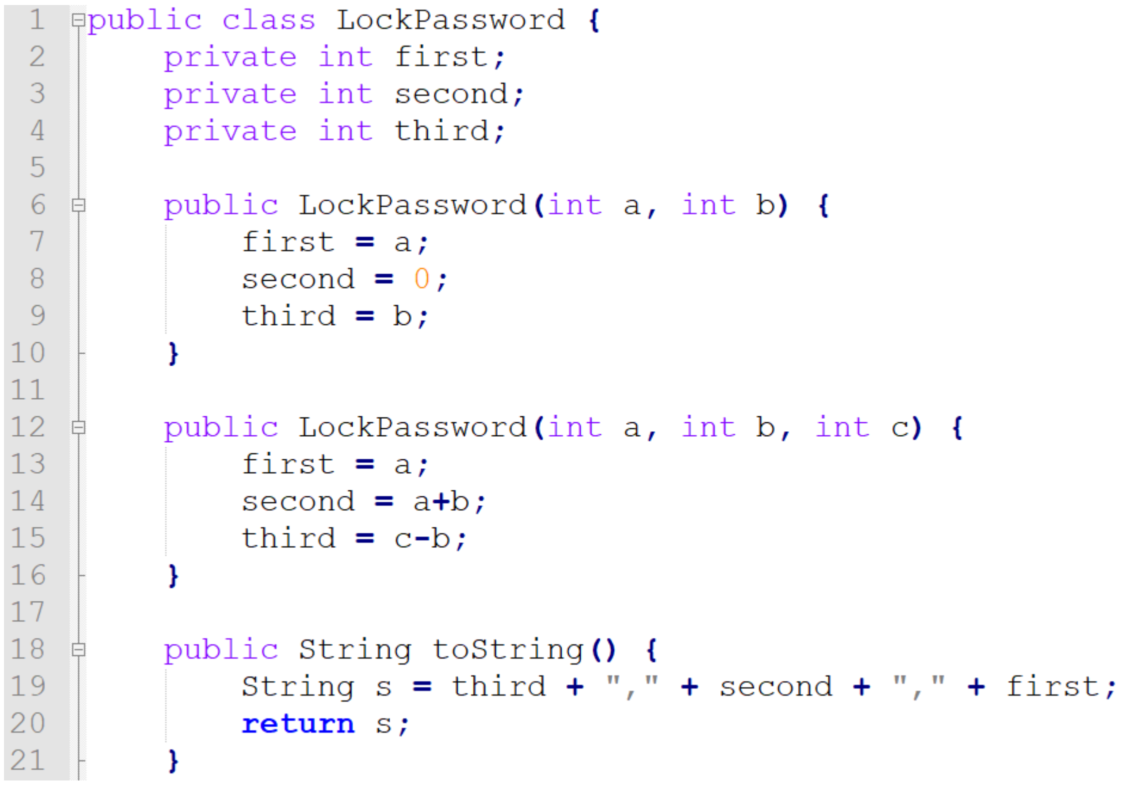
```
public interface Webpage {
    public String getURL();
}

public interface App {
    public double getPrice();
    public double getVersion();
    public boolean compatible(String phoneModel);
}

public interface Support {
    public String getSupportEmail();
    public String getSupportNumber();
    public void reportIssue(String issue);
    public String[] frequentlyAskedQuestions();
}

public interface Game {
    public String genre();
    public int numPlayers();
}
```

4. Given the 4 interfaces shown above, select which methods the following class must have in order to compile without error:

```
public class BigIdea implements Webpage, Support {
     //code here
}
```

[✓ **A. getURL()**]
B. getPrice()
C. getVersion()
D. compatible(String phoneModel)
[✓ **E. getSupportEmail()**]
[✓ **F. getSupportNumber()**]
[✓ **G. reportIssue(String issue)**]
[✓ **H. frequentlyAskedQuestions()**]
I. genre()
J. numPlayers()

Part 5: Objects (6 marks)



```java
public class LockPassword {
    private int first;
    private int second;
    private int third;

    public LockPassword(int a, int b) {
        first = a;
        second = 0;
        third = b;
    }

    public LockPassword(int a, int b, int c) {
        first = a;
        second = a+b;
        third = c-b;
    }

    public String toString() {
        String s = third + "," + second + "," + first;
        return s;
    }
```

5. Given the LockPassword class shown above, what is the output of the following lines of code:

   LockPassword aLock = new LockPassword(88, 29);
   System.out.println(aLock);

   Output: **29, 0, 88**

   LockPassword aLock = new LockPassword(1,4,7);
   System.out.println(aLock);

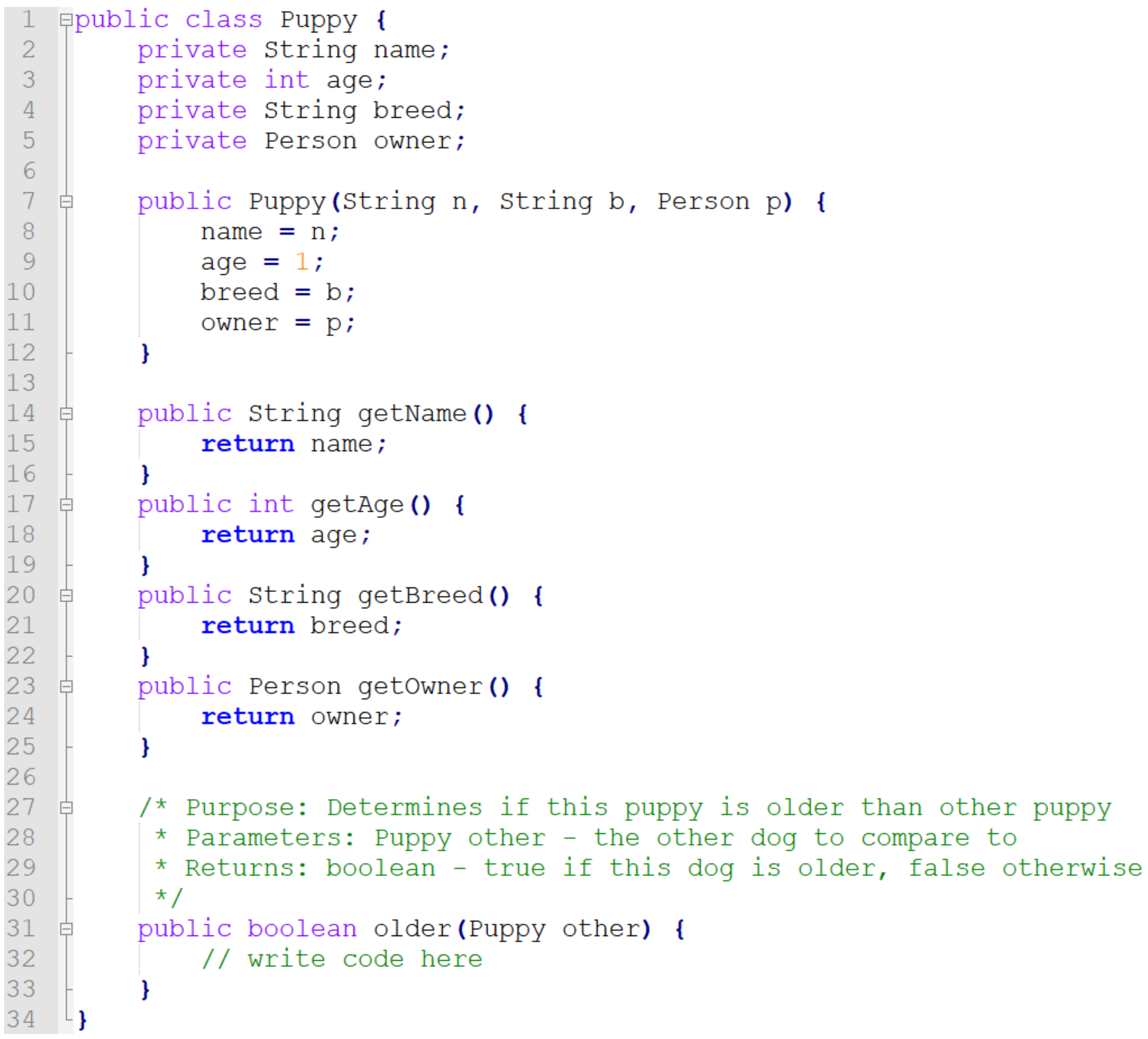
   Output: **3, 5, 1**

   Fill in the blank so that 3,2,1 is output.

   LockPassword aLock = new LockPassword(__**1, 1, 4**__);
   System.out.println(aLock);

Part 6: Code Writing (10 marks)

1. Given the Puppy class above, complete the older method, show beginning at line 31.



```
 1  public class Puppy {
 2      private String name;
 3      private int age;
 4      private String breed;
 5      private Person owner;
 6
 7      public Puppy(String n, String b, Person p) {
 8          name = n;
 9          age = 1;
10          breed = b;
11          owner = p;
12      }
13
14      public String getName() {
15          return name;
16      }
17      public int getAge() {
18          return age;
19      }
20      public String getBreed() {
21          return breed;
22      }
23      public Person getOwner() {
24          return owner;
25      }
26
27      /* Purpose: Determines if this puppy is older than other puppy
28       * Parameters: Puppy other - the other dog to compare to
29       * Returns: boolean - true if this dog is older, false otherwise
30       */
31      public boolean older(Puppy other) {
32          // write code here
33      }
34  }
```

Acceptable answers:

```
1) return this.age > other.getAge();

2) if (this.age > other.getAge()) {
        return true;
   } else {
        return false;
   }
```

2 marks – 1 mark deducted per mistake

```java
public class Person {
    String name;
    String phoneNumber;

    public Person(String n, String p) {
        name = n;
        phoneNumber = p;
    }

    public String getName() {
        return name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }
}
```

Given the Person class shown above, and the Puppy class shown in the previous exercise, complete the implementation of the getOwnerNumber shown below:

```java
/* Purpose: Get the phone number of the owner of the dog
 *          with the given name if is found in the dogs array
 * Parameters: Puppy[] dogs - the array of puppies
 *             String name - the name of the dog to search for
 * Returns: String - the phone number of the owner of the dog if a
 *                   dog with the name is found in dogs, "" otherwise
 */
public static String getOwnerNumber(Puppy[] dogs, String name) {
    // Implement this method
}
```

```java
String result = "";
for (int i = 0; i < dogs.length; i++) {
     if (dogs[i].getName().equals(name)) {
          result = dogs[i].getOwner().getPhoneNumber();
          break;
     }
}
return result;
```

8 marks: Parentheses shown to indicate how many were taken off if it was completely missing.
(1) loop: -1 per error, common errors: dogs.length
(2) if: -1 per error, common errors: dogs[i], .equals(), .getName()
(3) true case: -1 per error, common: dogs[i], getOwner(), getPhoneNumber()
else/false case: -2 if there unnecessary/incorrect code in an else case
(2) returns: -1 per error: missing returning correct phoneNumber or ""

… Left blank for scratch work…

END OF EXAM

Question	**Value**	**Mark**
Part 1: Array references	4	
Part 2: Syntax Errors	4	
Part 3: Testing and Debugging	6	
Part 4: Interfaces	4	
Part 5: Objects	6	
Part 6: Code Writing	10	
Total	**34**	