CSC 115

Midterm Exam:
Thursday, 23 May 2019

Name:_____(please print clearly!)

UVic ID number:_____

Signature:_____

Exam duration: 40 minutes

Instructor: Celina Berg

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. **Electronic devices are not permitted**.
- The marks assigned to each question and to each part of a question are printed within brackets. Partial marks are available.
- There are eight (8) pages in this document, including this cover page.
- Page 8 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.

Question 1 (5 marks) What is the output of the following program?

```
public class Q1 {
    public static void main(String args[]) {
        int[] a = {2, 3, 4};
        int v = 3;

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

        System.out.println();
        foo(a, v);

        System.out.println("a: ");
        for(int i=0; i<v; i++)
            System.out.println(a[i]);

        System.out.println();
    }

    public static int foo(int[] a, int v) {

        v--;

        for(int i=1; i<v; i++)
            a[i] *= v;

        System.out.println("a: ");
        for(int i=0; i<v; i++)
            System.out.println(a[i]);

        System.out.println();

        return v;
    }
}
```

Write only your final answer in this box for grading
Use the rest of the page for scratch work.

Question 2 (10 marks) Implement the following method according to the specification given.

```
/*
 * Purpose: Finds the longest occurring sequence length of
 *  the given value(val) in the given array (array)
 *
 * Parameters: int[] array, int val
 *
 * Examples: (longest sequences in examples are bolded)
 *  if a is {} then longest_sequence_of_val(a, 5) returns 0
 *  if a is {1,1,1,1} then longest_sequence_of_val(a, 1) returns 4
 *  if a is {1,1,1,1} then longest_sequence_of_val(a, 5) returns 0
 *  if a is {1,3,3,2,4,4,1,4,4,4,3} then
 *          longest_sequence_of_val(a, 5) returns 0
 *  if a is {1,3,3,2,4,4,1,4,4,4,3} then
 *          longest_sequence_of_val(a, 2) returns 1
 *  if a is {1,3,3,2,4,4,1,4,4,4,3} then
 *          longest_sequence_of_val(a, 3) returns 2
 *  if a is {1,3,3,2,4,4,1,4,4,4,3} then
 *          longest_sequence_of_val(a, 4) returns 3
 *
 * Returns: int - longest sequence length
 */
```

Question 3 (8 marks) What is the output of the following program? NOTE: this program is spread across 2 pages. Write your answer in the box provided on the next page.

```
public class PayRate {

    private int hrsPerWeek;
    private double wagePerHour;

    public PayRate() {
        hrsPerWeek = 0;
        wagePerHour = 0.0;
        System.out.println("A");
    }

    public PayRate(int hrsPerWeek, double wagePerHour) {
        this.hrsPerWeek = hrsPerWeek;
        this.wagePerHour = wagePerHour;
        System.out.println("B");
    }

    public void setHours(int hrsPerWeek) {
        this.hrsPerWeek = hrsPerWeek;
    }

    public int getHours() {
        return this.hrsPerWeek;
    }

    public void setWage(double wagePerHour) {
        this.wagePerHour = wagePerHour;
    }

    public double getWage() {
        return this.wagePerHour;
    }

    public int getSalary() {
        return (hrsPerWeek * (int) wagePerHour * 52);
    }

    public boolean equals(PayRate pr) {
        return this.getSalary() == pr.getSalary();
    }

    public String toString() {
        return hrsPerWeek + "*" + wagePerHour + " per week";
    }

    public void giveRaise(int percent) {
        // you will implement this in Question 4
    }
```

```
public static void main(String[] args) {

        PayRate prate1;
        PayRate prate2;
        System.out.println("C:");

        PayRate prate3 = new PayRate();
        System.out.println("D:" + prate3.getWage());

        PayRate prate4 = prate3;

        prate1 = new PayRate(20, 40.97);

        prate2 = new PayRate(40, 15.90);
        System.out.println("E:" + prate2);

        prate3 = prate2;
        prate3.setWage(20.27);

        System.out.println("F:" + prate1.getWage());
        System.out.println("G:" + prate2.getWage());
        System.out.println("H:" + prate3.getWage());
        System.out.println("I:" + prate4.getWage());

        System.out.println("J:" + prate1.equals(prate4));
        System.out.println("K:" + prate1.equals(prate3));
    }
}
```

Write **only** your final answer in this box.

Question 4 (5 marks)

Complete the implementation of the `giveRaise` method to be include in the `PayRate` class defined in Question 3 according to the following documentation:

```
/*
 * Purpose: increase this wagePerHour by percent %
 * Parameters: int percent
 *
 * Precondition: percent is >0
 *
 * Example:
 *  if percent is 3 then wagePerHour should increase by 3%
 *
 * Returns: nothing
 */
```

END OF EXAM

Question	Value	Mark
1	5	
2	10	
3	10	
4	5	
Total	**30**	