

German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher
Dr. Wael Aboul Saadat

May 28, 2016

CSEN 202: Introduction to Computer Programming Spring term 2015-2016 Final Exam

Bar Code

Instructions: Read carefully before proceeding.

- 1) Duration of the exam: 3 hours (180 minutes).
- 2) No books or other aids are permitted for this test.
- 3) This exam booklet contains 15 pages, including this one. Three extra sheets of scratch paper are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete.**
- 4) Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem or on the four extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets.**
- 5) When you are told that time is up, stop working on the test.

Good Luck!

Don't write anything below ; -)

Exercise	1	2	3	4	5	6	Σ
Possible Marks	18	12	12	12	12	34	100
Final Marks							

Exercise 1

(4+4+6+4=18 Marks)

- a) The `majority` method takes three boolean arguments and returns true if two (or more) of the arguments are true; and false otherwise. Complete two implementations of `majority` below by filling in the letter of one of the expressions below in each provided space. You may use each letter once, more than once, or not at all (but you may not use any other code).

1. false
2. true
3. x
4. y
5. z
6. x && y
7. x && z
8. y && z
9. x && y && z
10. x || y
11. x || z
12. y || z
13. x || y || z
14. count++
15. count--
16. count
17. count <= 1
18. count >= 2

```
public static boolean majority(boolean x, boolean y, boolean z) {
    int count = 0;
    if (____) ____;
    if (____) ____;
    if (____) ____;
    return ____;
}
```

Solution:

```
public static boolean majority(boolean x, boolean y, boolean z) {
    int count = 0;
    if (x) count++;
    if (y) count++;
    if (z) count++;
    return count >= 2;
}
```

```
public static boolean majority(boolean x, boolean y, boolean z) {
    if (____) return ____;
    if (____) return ____;
    if (____) return ____;
    return ____;
}
```

Solution:

```
public static boolean majority(boolean x, boolean y, boolean z) {
    if (x && y) return true;
    if (x && z) return true;
    if (y && z) return true;
    return false;
}
```

b) Consider the following Java code fragment:

```
int n = 100;
for (int i = 0; i < n; i++) {
    for (int j = 0; (j < n) && (j != i); j++) {
        System.out.println(i + "-" + j);
    }
}
```

The above code will print many statements. Some of the output statements are mentioned below. Which of the following will appear as output?

- 0-0
- 0-1
- 1-0
- 8-8
- 2-98
- 98-2
- 100-99

Solution:

```
1-0
98-2
```

c) Consider the following Java code fragment:

```
int[] a = { 1, 6, 5, 3, 0, 2, 4 };
int n = a.length;
int[] b = new int[n];
for (int i = 0; i < n; i++)
    b[a[i]] = i;
int[] c = new int[n];
for (int i = 0; i < n; i++)
    c[i] = a[b[i]];
```

What are the values of the elements in the arrays `b[]` and `c[]` after the above code fragment is executed with array

```
int[] a = {1,6,5,3,0,2,4}
```

Solution:

```
b[] 4 0 5 3 6 2 1
c[] 0 1 2 3 4 5 6
```

d) Consider the following class:

```
public class Prelim {
    public int i;
    public static int j;
    public Prelim(int a, int b) {
        i = a;
        j = b;
    }
    public String toString() {
        return i + " " + j;
    }
    public static void main(String[] args) {
        Prelim p1 = new Prelim(0, 1);
        Prelim p2 = new Prelim(2, 3);
        System.out.println(p1 + " " + p2);
        Prelim p3 = p1;
        p3.i = 4;
        System.out.println(p3.i == p1.i);
        Prelim[] p = new Prelim[4];
        p[0] = p1;
        p[1] = p2;
        System.out.println(p[2].i);

    }
}
```

What does running class `Prelim` print to the console? Justify your answers.

Solution:

The algorithm will print:

```
0 3 2 3
true
```

and gives a null pointer exception for the element at position 2 in the array `p` as the elements by default are set to `null`. Thus, the element at position 2 is `null` and doesn't have a value for `i`.

Exercise 2

(12 Marks)

Write a class called `Arithmetic` that uses three command-line arguments, namely two integers followed by an arithmetic operator (+, -, * or /) to perform the corresponding operation on the two integers and print the result. For example:

```
> java Arithmetic 3 2 +
3+2=5
```

```
> java Arithmetic 3 2 -
3-2=1
```

```
> java Arithmetic 3 2 /
3/2=1
```

You have to make sure that the user enters the correct number of arguments and one of the four operators. For example

```
> java Arithmetic 3 2 / 22
Usage: java Arithmetic int1 int2 op
```

```
> java Arithmetic 3 2 %
Error: invalid operator!"
```

Solution:

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

        if(args.length > 3 ||args.length < 3 )
            System.out.println("Usage: java Arithmetic int1 int2 op");
        else
        {
            if( args[2]!="+" || args[2]!="-" || args[2]!="*" || args[2]!="/" )
                System.out.println("Error: invalid operator!");
            else
            {
                int sum = 0;
                int op1 = Integer.parseInt(args[0]);
                int op2 = Integer.parseInt(args[1]);
                System.out.print(op1 + args[2] + op2 + " = ");
                switch(args[2].charAt(0))
                {
                    case '+':sum = op1 + op2;
                    System.out.println(sum);break;
                    case '-':sum = op1 - op2;
                    System.out.println(sum);break;
                    case '*':sum = op1 * op2;
                    System.out.println(sum);break;
                    case '/':sum = op1 / op2;
                    System.out.println(sum);break;
                    default: System.out.println("Error");
                }
            }
        }
    }
}
```

Exercise 3

(12 Marks)

Write a method that returns the greatest common factor (GCF) of two positive integers (the largest integer that divides them both. For example, the GCF of 98 and 70 is 14 because $98 = 2 \times 7 \times 7$ and $70 = 2 \times 5 \times 7$, and the greatest common factor of 98 and 85 is 1 because they have no common factors greater than 1.

Hint: To compute the GCF, define a public recursive static method `gcf()` that implements the following approach: the GCF of two numbers is their value if they are equal, and the GCF of the smaller one and the absolute value of their difference otherwise.

Solution:

```
public static int gcf(int p, int q) {  
    // base case  
    if (p == q) return p;  
  
    // call gcd with smaller and abs(difference)  
    int smaller = Math.min(p, q);  
    int difference = Math.abs(p - q);  
    return gcf(smaller, difference);  
}
```

Exercise 4

(12 Marks)

Write a recursive method `MinRec` that given an array of integers finds the smallest value in the given array.

Note that you are not allowed to use any additional arrays.

Running the following `main` method:

```
public static void main(String[] args) {  
  
    int[] a = {1,2,3,4};  
    int[] b = {5,2};  
    int[] c = {6,6};  
    System.out.println(MinRec(a));  
    System.out.println(MinRec(b));  
    System.out.println(MinRec(c));  
}
```

will display

```
1  
2  
6
```

Solution:

```
public static int MinRec(int []a){  
    return helper(a,1,a[0]);  
}  
  
public static int helper(int []a, int i , int min){  
    if(i == a.length)  
        return min;  
    else if(a[i] < min)  
    {  
        min = a[i];  
        return helper(a,++i, min);  
    }  
    else  
        return helper(a,++i,min);  
}
```

Exercise 5

(12 Marks)

A magic square of order n is an arrangement of $n \times n$ numbers, usually distinct integers, in a square, such that the n numbers in all rows, all columns, and both diagonals sum to the same constant.

Write a method `checkMagicSquare` that, given a two-dimensional array of numbers, returns whether it represents a magic square or not.

Example: For the following two-dimensional array

```
2 7 6
9 5 1
4 3 8
```

your program should return

This is a magic square with a sum 15

Example: For the following two-dimensional array

```
2 7 6
9 5 8
4 3 8
```

your program should return

This is not a magic square

Solution:

```
public static String checkSquare(int [][] a)
{
    int [] row = new int [a.length];
    int [] col = new int [a.length];
    int diag1 = 0, diag2 = 0;
    for (int i = 0; i < a.length; i++)
    {
        for(int j = 0; j < a[i].length; j++)
        {
            row[i] +=a[j][i];
            col[i] +=a[i][j];
            if(i == j)
                diag1 += a[i][j];
            if(i == a.length-1-i)
                diag2 += a[i][j];
        }
    }

    boolean f1 = true;
    boolean f2 = true;
    boolean f3 = diag1==diag2;

    for(int i = 0; i <= row.length/2 && f1; i++)
        f1 = row[i]==row[row.length-1-i];

    for(int i = 0; i <= col.length/2 && f2; i++)
        f2 = col[i]==col[col.length-1-i];
```



```
    if(f1 && f2 && f3)
        return "This is a magic square with sum "+diag1 ;
    else
        return "This is not a magic square";
}
```

Exercise 6

(34 Marks)

A date is defined as a particular day, month, and year.

a) Implement a class `Date` to define a date. Assume that an object of class `Date` has the following attributes:

- a day (where: 1 stands for Sunday, 7 stands for Saturday)
- a month (where: 1 stands for January, 12 stands for December)
- a year

The class should have an additional class variable which is an array of integers representing the number of days in every month. Assume that the first element correspond to the number of days in January and the last element corresponds to the number of days in December.

```
{31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}
```

The class `Date` should implement the following:

1. a constructor that takes three parameters as input
2. `public int getDay()`
returns the date's day.
3. `public int getMonth()`
returns the date's month.
4. `public int getYear()`
returns the date's year.
5. `public String toString()`
return a String of the form
DD/MM/YYYY
6. `public void updateDate()`
to add one day to the date. **Note:** You should handle all cases when the date is updated.

Solution Part a):**Solution:**

```
public class Date {

    int day;
    int month;
    int year;
    static int[] monthDays = {31,28,31,30,31,31,30,31,30,31,30,31};

    public Date(int day, int month, int year) {
        this.day = day;
        this.month = month;
        this.year = year;
    }

    public int getDay() {
        return day;
    }
    public int getMonth() {
        return month;
    }
    public int getYear() {
        return year;
    }

    public String toString()
    {
        String s = "";
        if(day<10)
            s+="0"+day+"/";
        else
            s+=day+"/";
        if(month<10)
            s+="0"+month+"/";
        else
            s+=month+"/";
        s+=year;
        return s;
    }

    public void updateDate()
    {
        if(day==monthDays[month-1])
        {
            day=1;
            if(month==12)
            {
                month = 1;
                year++;
            }
            else
                month++;
        }
        else
            ;
    }
}
```

```
        day++;  
    }  
}
```

b) **Hint: Read all the question then start to solve:**

Assume that a user would like to keep track of his/her vacation days. Implement a class called `VacationDays` that has two attributes

- `dateList` which is an array of `Date` objects
- `numEntries` which corresponds to the number of `Date` instances that are actually in the list (the remaining slots will be `null`).

The class `VacationDays` should implement the following:

1. a constructor that takes an array `d` of dates as input and initializes the corresponding instance variables. The array instance variable should be of the same length as the array `d`. Assume that the first `numEntries` of `d` are `Date` instances and the remaining elements are `null`. You are required to do deep cloning in the constructor. Do not copy references of the objects. **Hint:** You are required to determine the value of `numEntries`.
2. Implement a boolean method `inOrder()` which determines if the elements of `dateList` are in order.
3. Implement a `toString` method that displays the dates in a specific format. For example, if the list consists of two dates `new Date(11, 5, 2016)` and `new Date(14, 7, 2016)`:

```
You have vacation days in:  
day 11 in month 5 in year 2016  
day 14 in month 7 in year 2016
```

4. Implement a main method to test your classes.
 - Create an array `dates` of `Date` objects of size 10. The list should consist of at least three `Date` instances.
 - Create a `VacationDays` instance using the array `dates` and the corresponding constructor.
 - Check whether the dates are in order.
 - Display the information of the `VacationDays` object.
 - Update the date that appears in the first position of the array of days of the `VacationDays` object and display only this date using the corresponding methods from Part a).

Solution Part b)**Solution:**

```
public class VacationDays {

    Date []dateList;
    int numEntries;

    public VacationDays(Date []d)
    {
        dateList = new Date[d.length];
        for(int i=0;i<d.length&&d[i]!=null;i++)
        {
            dateList[i] = new
            Date(d[i].getDay(),d[i].getMonth(),d[i].getYear());
            numEntries++;
        }
    }

    public boolean inOrder()
    {
        for(int i=0;i<dateList.length-1 && dateList[i]!=null &&
        dateList[i+1]!=null ;i++)
        {
            if(dateList[i].getYear()>dateList[i+1].getYear())
            return false;
            if(dateList[i].getMonth()>dateList[i+1].getMonth() &&
            dateList[i].getYear()>=dateList[i+1].getYear())
            return false;
            if(dateList[i].getDay()>dateList[i+1].getDay() &&
            dateList[i].getMonth()>=dateList[i+1].getMonth())
            return false;
        }
        return true;
    }

    public String toString()
    {
        System.out .println("You have vacation days in:");
        for(int i=0;i<dateList.length&&dateList[i]!=null;i++)
        {
            System.out .println("day "+dateList[i].getDay()+
            " month"+dateList[i].getMonth()+
            " year "+dateList[i].getYear());
        }
        return "";
    }

    public static void main(String []args)
    {
        Date[]d = new Date[3];
        d[0] = new Date(11,5,2003);
        d[1] = new Date(1,3,2006);
        d[2] = new Date(13,6,2004);
        VacationDays v = new VacationDays(d);
    }
}
```

```
        System.out.println(v.inOrder());  
        System.out.println(v);  
        v.dateList[0].updateDate();  
        System.out.println(v.dateList[0]);  
    }  
}
```

Scratch paper

Scratch paper

Scratch paper
