**German University in Cairo**
**Faculty of Media Engineering and Technology**
**Prof. Dr. Slim Abdennadher**

June 13, 2009

# Introduction to Computer Programming Spring term 2009

Final Exam

**Bar Code**

**Instructions: Read carefully before proceeding.**

1) Duration of the exam: 3 hours (180 minutes).
2) (Non-programmable) Calculators are allowed.
3) No books or other aids are permitted for this test.
4) This exam booklet contains 13 pages including this one. Three extra sheets of scratch paper are attached and have to be kept attached. The exam consists of 6 exercises. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete**.
5) 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 three 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**.
6) 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 | $\sum$ |
|---|---|---|---|---|---|---|---|
| Possible Marks | 15 | 10 | 21 | 8 | 8 | 8 | 70 |
| Final Marks | | | | | | | |

**Exercise 1** (2+3+6+4=15 Marks)

a) Consider the following method:

```
public void f(int a, int b) {
    if (a < 0 || b < 0) {
       return;
    }
    int c = a + b;
    if (c < 0) {
       System.out.println("c is negative");
    }
}
```

Is it possible for the method to print `"c is negative"`? Why or why not? Explain briefly.

**Solution:**

Yes: The sum `a+b` could exceed the maximum positive integer that can be represented as an `int`.

b) Consider the following class:

```
class Point {
  private int x;
  private int y;


  public Point(int x, int y) {
     this.x = x;
     this.y = y; }

  public int getX() { return x; }

  public int getY() { return y; }

  public void setX(int x) { this.x = x;}

  public void setY(int y) { this.y = y;}
}
```

What is printed by the following statements?

```
Point p = new Point(17, 42);
Point q = p;
System.out.println(p.getX());  -->
p.setX(11);
System.out.println(p.getX());  -->
System.out.println(q.getX());  -->
p= new Point(9,29);
System.out.println(p.getX());  -->
System.out.println(q.getX());  -->
```

**Solution:**

```
Point p = new Point(17, 42);
Point q = p;
System.out.println(p.getX());  --> 17
p.setX(11);
System.out.println(p.getX());  --> 11
System.out.println(q.getX());  --> 11
p= new Point(9,29);
System.out.println(p.getX());  --> 9
System.out.println(q.getX());  --> 11
```

c) Given the following method

```
public static boolean mystery(int[] a, int[] b) {
     int i = 0;
     int j = 0;
     while(j < b.length && i < a.length) {
        if (b[j] == a[i]) {
          i++;
          j++;
        }
        else i++;
     }
     if (j == b.length)
        return true;
     else return false;
}
```

- Trace your method given the arrays

  ```
  a = {3,4,7,-1,10,22,9,-5}
  b = {4,-1,9,-5}
  ```

  **Solution:**

  ```
  true
  ```

- Trace your method given the arrays

  ```
  a = {3,4,7,-1,10,22,9,-5}
  b = {-1,9,4,-5}
  ```

  **Solution:**

  ```
  false
  ```

- What does the method do for any two given arrays `a` and `b` of integers?

  **Solution:**
  The method returns true if all elements of the array `b` occur in the array `a` with the same order they appear in `b`, otherwise the method returns `false`.

d) Consider the following program:

```
class Foo {

public static void main (String[] args) {
       double[] d = new double[Integer.parseInt(args[0])];
       for (int i = 0; i < d.length; i++) {
         d[i] = -i;
       }
       String s = new String("zippitydodah");
       if (s == (args[1])) {
          baz(d, true);
       }
       else {
         baz(d, false);
       }
       System.out.println("Length: " + d.length);
       for (int i = 0; i < d.length; i++) {
            System.out.println("d[" + i + "] = " + d[i]);
       }
} // main

public static void baz (double[] d, boolean x) {
     if (x) {
        d = new double[d.length / 2];
     }
     for (int i = 0; i < d.length; i++) {
         d[i] = i * 2;
     }
} // baz
} // class Foo
```

What is the output of this program if it is invoked like this? Justify your answer.

```
PROMPT> java Foo 4 zippitydodah
```

**Solution:**

```
Length: 4
d[0] = 0.0
d[1] = 2.0
d[2] = 4.0
d[3] = 6.0
```

**Exercise 2** (6+4=10 Marks)

a) Write a recursive method called `countVowels` that returns the number of vowels in a given String. Use the following helper method that checks whether a character is a vowel:

```
public static boolean isVowel(char c) {
      return "AaEeIiOoUu".indexOf(c) >= 0;
}
```

Example tests:

```
countVowels("") --> 0
countVowels("Four score and seven years") --> 9
countVowels("The quick brown fox") --> 5
countVowels("One if by Land - Two if by Sea") --> 8
```

b) Write a `main` method using the command-line argument to test the method `countVowels`.

For example

```
PROMPT> java  CountVowels GUC
1
PROMPT> java  CountVowels AUC
2
PROMPT> java  CountVowels
Sorry You did not enter any input.
```

**Solution:**

```
public class CountVowels {

        public static boolean isVowel(char c) {
             return "AaEeIiOoUu".indexOf(c) >= 0;
        }

        public static int countVowels(String s) {
             if (s.equals("")) {
               return 0;
             }
             char first = s.charAt(0);
             int total = 0;
             if (isVowel(first)) {
                 total++;
             }
             total += countVowels(s.substring(1));
             return total;
        }

        public static void main(String[] args) {
           if(args.length == 0)
               System.out.println("Sorry You did not enter any input.");
           else
               System.out.println(countVowels(args[0]));
        }

}
```

**Exercise 3** (12+9=21 Marks)

a) An `Initials` object is defined by the first letter of the first name and the first letter of the last name.

1. Write a class to define the `Initials` object.
2. Write a constructor that takes two characters and initializes the instance variables of the class.
3. Write a second constructor that takes the first- and last name of a person and create an object of type `Initials`. You have to use the first constructor to define the second constructor.
4. Write two setter to set the initials and two getter methods to get the initials.
5. Write a `toString` method to display the initials.
6. Write a static method that compares two initials. It should return true if the initials are identical otherwise the method should return false.
7. Write a `main` method to test all the methods defined in the class `Initals`.

**Solution:**

```
class Initials{
        char firstInit;
        char lastInit;

        public Initials(char fI, char lI){
                firstInit = fI;
                lastInit = lI;
        }

        public Initials(String firstName, String lastName){
                this(firstName.charAt(0), lastName.charAt(0));
        }

        public void setFirst(char fI){
                 firstInit = fI;
        }

        public void setLast(char lI){
                 lastInit = lI;
        }

        public char getFirst(){
              return firstInit;
        }

        public char getLast(){
               return lastInit;
        }

        public String toString(){
               return "Initials:" + firstInit + ", " + lastInit;
        }

        public static boolean compare(Initials I1, Initials I2){
               if(I1.getFirst() == I2.getFirst() && I1.getLast() == I2.getLast()){
                  return true;
                }
                else{
                  return false;
                }
        }

        public static void main(String [] args){
                Initials I1 = new Initials('S', 'A');
                Initials I2 = new Initials("Kobe", "Bryant");

                System.out.println(I1.getFirst());
                System.out.println(I2.getLast());
                System.out.println(I1);
                System.out.println(I2);
                System.out.println(compare(I1, I2));
        }
}
```

b) Given the following class that defines the object `Person`:

```
public class Person {
    String firstname;
    String lastname;
    char gender;
}
```

1. Define a class called `Persons` that is defined by a list of `Person` objects and their corresponding list of `Initials`.
2. Write a constructor that initializes the instance variables given the list of objects of type `Person`.
3. Write a method that returns an array of arrays. In the first array, the initials of male persons will be stored and in the second array the initials of female persons will be stored.
4. Write a `main` method to test the methods above. In the main method the content of the two dimensional array should be displayed in a tabular form. Assume that there is a `toString` method in the `Person` class.

**Solution:**

```
class Person{
        String firstname;
        String lastname;
        char gender;

public Person(String fN, String lN, char g){
      firstname = fN;
      lastname = lN;
      gender = g;
}
}



class Persons{
        Person[] P;
        Initials[] I;

public Persons(Person [] pArr){
       P = new Person[pArr.length];
       I = new Initials[pArr.length];

       for(int i = 0; i < pArr.length; i++){
          P[i] = new Person(pArr[i].firstname, pArr[i].lastname, pArr[i].gender);
          I[i] = new Initials(pArr[i].firstname.charAt(0), pArr[i].lastname.charAt(0));
       }
}

public Initials [][] splitGender(){
        Initials[][] I = new Initials[P.length][2];

        int m = 0;
        int f = 0;
        int i = 0;
        while (i < P.length){
             if (P[i].gender == 'm'){
                 I[m][0] = new Initials(P[i].firstname.charAt(0),
                                        P[i].lastname.charAt(0));
                 m++;
             }
             else if(P[i].gender == 'f'){
                    I[f][1] = new Initials(P[i].firstname.charAt(0),
                                           P[i].lastname.charAt(0));
                    f++;
              }
          i++;
          }
          return I;
}

public static void main(String[] args){

    Person[] P = new Person[] {new Person("Michael", "Jordan", 'm'),
                               new Person("Becky", "Hammon", 'f')};
    Persons People = new Persons(P);
    Initials[][] I = People.splitGender();
    System.out.println("Males" + "            " + "Females");
    for(int i = 0; i < I.length; i++){
        for(int j = 0; j < I[i].length; j++){
           if(I[i][j] != null) {System.out.print(I[i][j] + "     ");}
        }
        System.out.println("");
     }
}
}
```

**Exercise 4** (8 Marks)

Write a method `prefix` that, given a string `s`, returns an array of strings containing all prefixes of `s` in increasing order (`s` included). For example, given the string `"abcdef"`, the method returns the array:

```
{"a", "ab", "abc", "abcd", "abcde", "abcdef"}
```

**Solution:**

```
public static String[] prefix(String s) {
     int n  = s.length();
     String[] a = new String[n];
     for(int i = 0; i < n; i++)
        a[i] = s.substring(0,i+1);
     return a;
}
```

**Exercise 5** (8 Marks)

Write a method that, given an array `a` of strings and an integer `k` (`k` $\geq 2$), returns `true` if in `a` the-re exist at least `k` strings of equal length, otherwise the method returns `false`. For example, given `a = {"GUC", "SLIM", "PHYSICS", "CSEN", "", "GOOD"}` and `k=3`, the method returns `true`.

**Solution:**

```
public static boolean occurence(String[] a, int k) {
    int i = 0;
    int count = 1;
    while (i < a.length)  {
        int j = i+1;
        while (j<a.length) {
           if (a[i].length() == a[j].length()){
                 count++;
           }
         if (count >= k) {
            return true;
         }
        j++;
        }
     i++;
     count = 1;
    }
    return false;
}
```

**Exercise 6** (8 Marks)

Create a method `checkFirstCharacter` that takes a two-dimensional array of Strings as parameter and returns `true` if all elements of any array start with the same character otherwise the method should return false. For example, suppose the array is of the following form

```
{{"ali", "ab", "an", "array"}, {"oops", "opla", "oh"}}
```

is passed to the method, the method would return `true`.

If the array is of the following form:

```
{{"bye", "aurevoir", "salam"}, {"oops", "opla", "oh"}}
```

the method would return `false`.

**Solution:**

```
public static boolean checkFirstCharacter(String[][] a) {
int i = 0;
int j = 0;
boolean p = true;
for (i=0; i < a.length; i++) {
   j = 0;
   p = true;
   while(j < a[i].length-1 && p) {
      if (a[i][j].charAt(0) == a[i][j+1].charAt(0)) {
           j++;
      }
      else p = false;
   }
}
return p;
}
```

**Extra Sheet**

**Extra Sheet**

**Extra Sheet**