German University in Cairo
Computer Science Department
Prof. Dr. Slim Abdennadher

June 5, 2004

Introduction to Computer Programming
Spring term 2004

Final Exam

Bar Code

Instructions: Read carefully before proceeding.

1) The duration of the exam: **3 hours**
2) No books or other aids are permitted for this test.
3) 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 extra sheets.
4) This exam booklet contains 19 pages, including this one. Four extra sheets of scratch paper are attached and have to be kept attached.
5) The exam consists of 12 exercises. Exercise 12 is a **bonus** one.
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	7	8	9	10	11	**12**	$\sum$
Marks	10	9	5	6	10	8	10	12	12	10	8	**8**	100
Final Marks													

Exercise 1 (10 Marks)

1) `"0"` is an example of what kind of constant?

- ☐ `int`
- ☐ `char`
- ☐ `boolean`
- ☐ [✓ `String`]
- ☐ other

2) Assume that `x` and `y` are integer variables and the following nested if statement.

```
if (x > 3) {
  if (x <= 5) y = 1;
  else if (x != 6) y = 2;
       else y = 3;
} else y = 4;
```

If the variable `y` has a value of `3` after executing the above program fragment, then what do you know about the variable `x`?

- ☐ `x!=6`
- ☐ [✓ `x==6`]
- ☐ `x==4`
- ☐ `x!=4`
- ☐ `x>3 && x<=4`

3) How many question marks will be printed by the following code fragment?

```
for (int i = -10; i <= 10; i = i + 3) ;
  System.out.println("?");
```

- ☐ [✓ 1]
- ☐ 2
- ☐ 3
- ☐ 6

4) What is the output produced by the following code when embedded in a complete program?

```
int i;
for (i = 0; i < 3; i = i + 1)
  System.out.print(1);
System.out.print(0);
```

- ☐ `0`
- ☐ [✓ `1110`]
- ☐ `11110`
- ☐ `101010`
- ☐ `10101010`

5) The following program segment is incorrect. Why?

```
int q = 0;
if (q = 1)
   q = 2;
```

- ☐ The name `q` is not a valid name for a Java variable
- ☐ You cannot assign 2 to a Java `int` variable.
- ☐ [✓ The `if` condition does not evaluate to true or false.]

6) Given the following method:

```
public int tryPrimitives(int x, int y) {
       x = x + 10;
       y = y + 10;
       return x;
    }
```

what are the values of `p`, `q` and `r` after the following code executes?

```
public static void main(String[] args) {
       int p = 1;
       int q = 2;
       int r = 5;
       r = tryPrimitives(p,q);
}
```

- ☐ `p=1, q=2, r=5`
- ☐ `p=11, q=12, r=5`
- ☐ [✓ `p=1, q=2, r=11`]
- ☐ `p=11, q=12, r=11`

7) You compile and run this program. What is the output?

```
class Account {

  int balance;
  Account (int n) { balance = n; }
  Account() { this(20); }
  int getBalance() { return balance + 30; }
  public static void main(String[] args) {
    Account e = new Account();
    Account f = new Account(10);
    System.out.println(e.getBalance() + f.getBalance());
  }
}
```

- ☐ `10`
- ☐ `30`
- ☐ `60`
- ☐ [✓ `90`]
- ☐ `Other`

Exercise 2 (9 Marks)

Consider the following definition:

```
int    one = 1;
double two = 2.0;
String four = "4";
```

Specify the type **and** the value of the following expressions or the reason why there is no value.

a) `(one + two) + four`

Solution:

`3.04`: String

b) `one + (two + four)`

Solution:

`12.04`: String

c) `2 <= two`

Solution:

`true`: Boolean

d) `2 = two`

Solution:

Syntax error: Variable name cannot be a number.

e) `(one / (one + one)) * two > 0`

Solution:

`false`: Boolean

f) `(one - one) / (one - one)`

Solution:

Run time error:

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
```

Exercise 3 (1+4=5 Marks)

Given the following iterative program to calculate the factorial of a number n.

```
int fac = 1;
int i;
for (i=1; i<=n; i++)
{
    fac = i * fac;
}
System.out.println(n +  " Factorial = " + fac);
```

a) Why is the program correct for `n=0`. Note that $0! = 1$.

Solution:

Since the condition `1<=0` is false, the body of the loop will not be executed and the result is `fac = 1`.

b) Rewrite the program using a while loop with the same behaviour as the program above.

Solution:

```
int fac = 1;
int i = 1;
while (i<=n)
{
    fac = i * fac;
    i++;
}
System.out.println(n +  " Factorial = " + fac);
```

Exercise 4 (6 Marks)

a) A certain application defines a static method with the following header:

```
public static double myMethod(double xVal)
```

Mark the following invocations of `myMethod` as correct or incorrect. Indicate why an invocation is **incorrect**.

- `System.out.println("Here it is: " + myMethod(7.0 * 3.5));`

  Solution:
  Correct

- `double  answer = myMethod("5.77");`

  Solution:
  Incorrect; the actual parameter cannot be a String

-
  ```
  double x = 3.14;
  ...
  myMethod(x);
  ```

  Solution:
  Correct

-
  ```
  double x = 3.14;
  System.out.println("Here it is: " + x.myMethod());
  ```

  Solution:
  Incorrect; the method is a static method.

b) The following is a class definition. Identify:

- The class method
- The instance method
- The class variable
- The instance variable

```
class Exam {
  private float grade;
  static float Max = 4.0f;
  static boolean isValid (float ckGrade) {
       if (ckGrade >= 0.0f && ckGrade <=Max) return true;
       else retrun false;
  }
  void setGrade(float newGrade) {
       grade = newGrade;
  }
}
```

Solution:

The `isValid` method is the class method, the `setGarde` is the instance method, `Max` is the class variable and `grade` is the instance variable.

Exercise 5 (10 Marks)

a) Given the following program

```
public static int whatIsThat(int n)
{
    if (n == 1)
       {return 1;}
    else
       {return n + whatIsThat(n-1);}
}
```

- What is the result of `whatIsThat(5)`? Show your workout.

  Solution:

```
whatIsThat(5) = 5 + whatIsThat(4)
              = 5 + (4 + whatIsThat(3))
              = 5 + (4 + (3 + whatIsThat(2)))
              = 5 + (4 + (3 + (2 + whatIsThat(1))))
              = 5 + (4 + (3 + (2 + 1)))
              = 15
```

- What is the output of the program for any positive integer number `n`? Give the general formula.

  Solution:

$$\sum_{i=1}^{n} i = n + (n-1) + (n-2) + \ldots + 1 = \frac{(n+1) \times n}{2}$$

b) Given the following program

```
public static String mystery(int n)
{
  if(n == 0)
    return "0";
  else if(n == 1)
    return "" + (n % 2);
  else
    return mystery(n / 2) + (n % 2);
}
```

- What is the result of `mystery(10)`? Show your workout.

  Solution:

```
mystery(10) = mystery(5) + 0
            = (mystery(2) + 1) + 0
            = ((mystery(1) + 0) + 1) + 0
            = (((mystery(0) + 1) + 0) + 1) + 0
            = ((("" + 1) + 0) + 1) + 0
            = 1010
```

- What does this program compute for any positive integer number `n`?

  Solution:

  `mystery` calculates the binary representation of a decimal number `n`.

Exercise 6 (8 Marks)

The binomial coefficient $\binom{n}{k}$ is the number of ways of picking k unordered outcomes from n possibilities, also known as a combination or combinatorial number.

The binomial coefficient is defined recursively as follows:

$$\binom{n}{k} = \begin{cases} 1 & : \text{ if } k = 0 \\ 1 & : \text{ if } n = k \\ \binom{n-1}{k} + \binom{n-1}{k-1} & : \text{ otherwise} \end{cases}$$

Write a recursive method to calculate the binomial coefficient.

Write a `main` method that will allow the user to enter the actual parameters of the binomial coefficient method. Use either the BufferReader or the command line arguments as input mechanism.

```
public static void main(String[] args) throws IOException {
        BufferedReader stdin=new BufferedReader(new InputStreamReader(System.in));
        ....
```

Solution:

```
public class Binom
{

  public static long choose(int n, int k)
  {
    long answer;             // Hold the answer
    if(k == 0 || n == k)
      answer = 1;
    else
      answer = choose(n-1, k) + choose(n-1, k-1);

    return answer;
  }


  public static void main(String args[])
  {
    int n = Integer.parseInt(args[0]);                // size of the set
    int k = Integer.parseInt(args[1]);;                // number of objects

    System.out.println("There are " + choose(n, k) + " ways to choose "
           + k + " objects out of " + n);
  }
}
```

Exercise 7 (3+2+5=10 Marks)

Given the following program:

```
public class Mobile
{
    public String model;
    public int number;

    public Mobile(String model)
    {   this.model = model;   }

    public static void main(String[] args)
    {   // I buy a new "Nokia" mobile and I can choose the number
        Mobile myMobile = new Mobile("Nokia");
        myMobile.number = 223344;
        // You buy a mobile of the same model, but you would like to have a different number:
        Mobile yourMobile = myMobile;
        yourMobile.number = 556677;
    }
}
```

a) What is the value of `myMobile.number` after the execution of the program? Explain the behaviour.

Solution:

The value of `myMobile.number` is `556677`, because `yourMobile` object has the same address as `myMobile`, thus values are reflected in both objects.

b) This effect is undesirable. How should the program be changed in order to avoid this effect and to have the result described in the comments?

Solution:

```
Mobile yourMobile = new Mobile(myMobile.model);
yourMobile.number = 556677;
```

c) The mobile company noticed that it is not a good idea that the customers choose their phone numbers. Change the class `Mobile` in such a way that every new mobile will be assigned an unused number (starting by 100001).

Solution:

Add an class variable `possibleNumber` and change the constructor as follows:

```
public class Mobile
{
    public String model;
    public int number;
    public static int possibleNumber = 100001;

    public Mobile(String model)
    {   this.model = model;
        this.number = possibleNumber;
        possibleNumber++;  }
}
```

Exercise 8 (12 Marks)

Design and implement a class `SimpleEmployee` with the following attributes:

- the employee's name (`String`)
- annual salary (`double`)

a) Define a class `SimpleEmployee` with the attributes defined above.

b) Augment your class with a constructor that initializes `SimpleEmployee` with an employee's name and annual salary.

c) Augment your class with the following methods:

- `getName()` to get the name of the employee.
- `getAnnualSalary()` to get the annual salary.
- `getMonthlySalary()` to calculate the monthly salary.
- `giveRaise(double p)` to apply a raise `p` as a percentage increase in annual salary.
- `Compare(SimpleEmployee e)` returns `true` if and only if the annual salary of the employee on which the method will be invoked is larger than the salary of the employee `e`.
- `display()` returns a `String` that provides a textual representation of the object (name and annual salary) .

d) Augment your class with a class variable to keep a track of every instance of the class `SimpleEmployee`.

e) Augment your class with a method that returns the total number of created employees.

f) Augment your class with a method `main` that constructs at least two employees and tests all defined methods.

Solution:

```
public class SimpleEmployee{
    String name;
    double annualSalary;
    static int number = 0;


    public SimpleEmployee(String theName, double theAnnualSalary){
           name = theName;
           annualSalary = theAnnualSalary;
           number++;
    }

    public String getName() {
           return name;
    }

    public double getAnnualSalary() {
           return annualSalary;
    }

    public double getMonthlySalary() {
           return annualSalary/12;
    }

    public void giveRaise(double p) {
           annualSalary =  annualSalary + ((p/100) * annualSalary);
    }

    public boolean compare(SimpleEmployee e) {
           if (annualSalary > e.annualSalary)
              return true;
           else
              return false;
    }

    public String display() {
           return  "The name  of the employee "  + name +
                   ". His annual salary  is " + annualSalary;
    }


    public static int total() {
           return number;
    }

    public static void main(String args[]){

    SimpleEmployee s1 = new SimpleEmployee("Mona", 120000.0);
    SimpleEmployee s2 = new SimpleEmployee("Ahmed", 80000.0);

    System.out.println(s1.display());
    System.out.println(s1.getName());
    System.out.println(s1.getAnnualSalary());
    System.out.println(s1.getMonthlySalary());

    System.out.println(s2.display());
    s2.giveRaise(15);
    System.out.println(s2.display());
    System.out.println(s1.compare(s2));
    }
}
```

Exercise 9 (12 Marks)

Given an array of integers of length 100. Write a Java program that initializes each element to be twice its position in the array and then calculates the sum of the squares of the elements.

The program should perform the following operations:

a) Create a new array of integers of length 100.

b) Print the array elements before initialization.

c) Initialize each element to be twice it's position in the array (using a loop).

d) Print the array elements after initialization.

e) Calculate and print the sum of the squares of the elements.

Solution:

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

int result = 0;

// create and instantiate a new array of integers
int numbers[] = new int[100];

System.out.println("The array elements BEFORE initialization are: ");

for (int i = 0; i < numbers.length; i++){
    System.out.print(numbers[i] + "\t");
} // end of for loop

System.out.print("\n");

// By definition the array elements are initialized to zero. Now we
// will initalize them to values we will use in our program.
// Initalize each element to be twice it's position in the array
for (int i = 0; i < numbers.length; i++){
    numbers[i] = i * 2;
}// end of for loop

System.out.println("the array elements after initialization are: ");

for (int i = 0; i < numbers.length; i++){
    System.out.print(numbers[i] + "\t");
} // end of for loop

System.out.print("\n");

// Now calculate the sum of the squares of the elements in numbers
for (int i = 0; i < numbers.length; i++) {
    result += numbers[i] * numbers[i];
} // end of for loop

System.out.println("The result is: " + result);


    } // end of main method
} // end of ArrayInit Class
```

Exercise 10 (10 Marks)

Merging 2 sorted arrays of integers of the same length (`A1,A2,...,An`) and (`B1,B2,....,Bn`), which are sorted in ascending order, produces a third array (`C1,C2,...,C2n`) which is also sorted. Consider the following example:

Array A:	2	11	12	34				
Array B:	4	7	25	45				
Array C:	2	4	7	11	12	25	34	45

a) Write a Java method `mergeArrays` that will merge 2 sorted arrays and returns the sorted merged array as result.

b) Write a `main` method to test your program and to display the elements of the result.

Solution:

There are many algorithms to merge two arrays.

- The simplest algorithm is to construct one array with a length equal to the lengths of the two arrays and then to start initalizing the elements of the array starting by the elements of the first array followed by the elements of the second array. Finally, sort the resulting array using for example the selection sort algorithm.
- In the following, a second algorithm is presented. In this case, we have to ensure that the subscript is within the bounds of the array.

```
class mergeArrays{
      public static int[] mergeArrays(int[] a, int[] b) {
                int[] c=new int[a.length+b.length];
                int i=0,j=0;
                for(int k=0;k<c.length;k++)
                {
                if (i<a.length && j<b.length)
                {
                   if (a[i]<=b[j])
                   {
                     c[k]=a[i];
                     if(i<a.length) i++;
                   }
                     else{
                         c[k]=b[j];
                         if(i<b.length) j++;
                         }
                }
                else
                {
                if (i<a.length)
                {
                   c[k]=a[i];
                   if(i<a.length) i++;
                }
                else{
                    c[k]=b[j];
                    if(j<b.length) j++;
                    }
                }
                }
                return c;
        }
        public static void main(String[] args){
```

```
                    int[] b = {1,3,4,5};
                    int[] a = {2,6,7,8};
                    int[] c = mergeArrays(a,b);
                    for(int k=0;k<c.length;k++)
                    {
                       System.out.print(c[k] + "\t");
                    }
            }
    }
```

Exercise 11 (8 Marks)

Write a Java method `disjointArrays` that takes two arrays of integers as parameters and **returns** `true` if and if the two arrays are disjoint, otherwise the method should return `false`. Two arrays are *disjoint* if and only if they do not have common elements. For example:

- `{1,3,4,5}` and `{2,4,6}` are not disjoint.
- `{1,3,4,5}` and `{2,6,8}` are disjoint.

Write a `main` method to test your program with the examples above. The `main` method should display either `The arrays are not disjoint` or `The arrays are disjoint`.

Solution:

```
class disjointArrays
{
    public static boolean disjointArrays(int[] a, int[] b) {

           boolean disjoint = true;
           int i=0, j=0;
           for (i=0; i<a.length; i++) {
               for (j=0; j<b.length; j++) {
                   if (a[i]==b[j])
                     { disjoint = false; }
       }
           }
           return disjoint;
    }


    public static void main (String[] args) {
        int[] a = {1,3,4,5};
        int[] b = {2,6};

        if (disjointArrays(a,b))
           System.out.println("The arrays are disjoint");
        else
           System.out.println("The arrays are not disjoint");
    }
}
```

Exercise 12 (8 Marks)

BONUS

The hyperexponent of two numbers x and y is defined as follows:

$$\text{hyperexponent}(x, y) = x^{\underbrace{x^{\cdots^{x}}}_{y\text{-times}}}$$

For example, the hyperexponent of 2 and 3 is

$$\begin{aligned}\text{hyperexponent}(2,3) &= 2^{2^{2^{2}}} = (((2)^2)^2)^2 \\ \text{hyperexponent}(5,4) &= 5^{5^{5^{5^{5}}}} = ((((5)^5)^5)^5)^5\end{aligned}$$

a) Write an iterative method to calculate the hyperexponent of two numbers. **Hint:** Use `Math.pow(n,m)` to calculate n^m.

b) Write a recursive method to calculate the hyperexponent of two numbers. **Hint:** Try first to find the recursive formula for hyperexponent(x, y). Use `Math.pow(n,m)` to calculate n^m.

Solution:

```
class hyperexponent
{
    public static double  hyperexponentRec(double x,  double y) {
           if (y==1.0)
              return Math.pow(x,x);
           else
              return Math.pow(hyperexponentRec(x,y-1),x);
    }

    public static double  hyperexponentIter(double x,  double y) {
        double result = x;
        while (y>=1.0) {
            result = Math.pow(result,x);
            y--;
        }
        return result;
    }

    public static void main (String[] args) {
      System.out.println(hyperexponentRec(2.0,3.0));
      System.out.println(hyperexponentIter(2.0,3.0));
    }
}
```