

Introduction to Computer Science Spring term 2010 Midterm Exam

Bar Code

Instructions: Read carefully before proceeding.

- 1) Duration of the exam: 2 hours (120 minutes).
- 2) (Non-programmable) Calculators are allowed.
- 3) No books or other aids are permitted for this test.
- 4) This exam booklet contains 12 pages (5 exercises), including this one. Four 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.**
- 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	Σ
Possible Marks	6	10	8	10	22	56
Final Marks						

Exercise 1

(3+3=6 Marks)

- a) What will be displayed on the screen when the below program is executed? Briefly explain your answer.

```
public class DanglingElse
{
    public static void main(String[] args)
    {
        int a=3,b=4;
        if(a>2)
            if(b==6)
                System.out.println("artichoke");
        else
            System.out.println("asparagus");
        System.out.println("broccoli");
    }
}
```

Solution:

```
asparagus
broccoli
```

The `else` statement belongs to the second `if` statement. The first `if` does not have an `else` part.

- b) What will be displayed on the screen when the below program is executed? Briefly explain your answer.

```
public class Deceive {
    public static void main(String[] args) {
        int n, result, i;
        n = 5;
        i = 0;
        result = 1;
        while (i < n)
            i = i + 1;
            result = result * i;

        System.out.println(result);
    }
}
```

Solution:

```
5
```

The statement `result = result * i;` does not belong to the body of the `while` loop. The body of the `while` loop consist of one statement since the curly brackets are missing.

Exercise 2

(5+5=10 Marks)

Given the following program that deals only with positive integers:

```
public static void main(String[] args) {
    int n = 8;
    if (n>=10 || n==7) {
        System.out.println("The number is 7,");
        System.out.println(" or is outside the range 1 to 9.");
    }
    else
        if (n==1)
            System.out.println("The number is 1.");
        else
            if (n%2 == 0) {
                System.out.println("The number is 2, 4, or 8.");
                System.out.println("(That's a power of 2!)");
            }
            else {
                if (n%3 == 0) {
                    System.out.println("The number is 3, 6, or 9.");
                    System.out.println("(That's a multiple of 3!)");
                }
                else
                    System.out.println("The number is 5.");
            }
    }
}
```

- a) Transform the program above to a program that uses **only** a switch statement.

Solution:

In the question, there was a mistake in the condition ($n\%2 == 0$). For $n=6$, the first condition that holds is ($n\%2 == 0$). In the correction, we will take that into consideration. According to the meaning of the code above, the corresponding program is:

```
public static void main(String[] args) {
    int n = 8;
    switch (n) {
        case 1:
            System.out.println("The number is 1.");
            break;
        case 2:
        case 4:
        case 8:
            System.out.println("The number is 2, 4, or 8.");
            System.out.println("(That's a power of 2!)");
            break;
        case 3:
        case 6:
        case 9:
            System.out.println("The number is 3, 6, or 9.");
            System.out.println("(That's a multiple of 3!)");
            break;
        case 5:
            System.out.println("The number is 5.");
            break;
        default:
            System.out.println("The number is 7,");
            System.out.println(" or is outside the range 1 to 9.");
    }
}
```

```
    }  
}
```

- b) Transform the program above to a program that uses **only** conditional operators.

Solution:

```
public static void main(String[] args) {  
    int n = 8;  
    System.out.println(  
        (n>=10 || n==7)?"The number is 7 \n or is outside the range 1 to 9."  
        (n==1)?"The number is 1."  
        (n%2==0)?"The number is 2, 4, or 8. \n(That's a power of 2!)":  
        (n%3 == 0)?"The number is 3, 6, or 9.\n(That's a multiple of 3!)":  
        "The number is 5.");  
}
```

Exercise 3

(8 Marks)

Write a method that takes as an argument n an even integer and displays the following shape

- for $n=6$:

```
* * *
 * * *
 ***
  *
 ***
 * * *
* * *
```

- for $n=8$

```
* * *
 * * *
 * * *
 ***
  *
 ***
 * * *
* * *
* * *
```

Solution:

```
public class Shape {
    public static void shape(int n) {
        for (int i = 0; i <= n; i++) {
            for (int j = 0; j <= n; j++) {
                if ((j == i) || (j == n-i) || (j == n/2)) {
                    System.out.print("*");
                } else {
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

Exercise 4

(10 Marks)

Consider the following program, saved in a file called `ContainsAtIndex.java`:

```

1 public class ContainsAtIndex {
2     public static void main(String[] args) {
3         Scanner keyboard = new Scanner(System.in);
4         String s;
5         String singleChar;
6         char c;
7         int i;
8
9         System.out.print("Enter a string: ");
10        s = keyboard.nextLine();
11
12        System.out.print("Enter a character: ");
13        singleChar = keyboard.nextLine();
14        // Assumes that the user will not enter an empty line
15        c = singleChar.charAt(0);
16
17        found = false;
18        while (i < s.length() && found) {
19            if (s.charAt(i) == c) {
20                found = true;
21            }
22            i = i + 1;
23        }
24
25        if (found) {
26            System.out.println("The first occurrence of '" + c +
27                "' in \"" + s + "\" is a position " + i);
28        } else {
29            System.out.println("There are no occurrences of '" + c +
30                "' in \"" + s + "\"");
31        }
32    }
33 }

```

The above program is designed to check whether a `String` entered by the user contains a character entered by the user, and if it does, to display the position of the first occurrence of the character in the `String`. However, there are 5 errors in the above program. Find all the errors and list them. For each error you list, you **MUST** must include the number of the line where the error occurs, the type of error (syntactic or semantic) and a description of the error. Do not list more than 5 errors, as you will be penalized for every "error" in excess of 5 that you list. Note that the line numbers to the left of the above program are included solely to help you make it easier for you to list the line numbers where errors occur; they are not part of the actual program.

Solution:

a) Syntactical error:

```
import java.util.*;
```

is missing.

b) Syntactical error:

```
17    found = false;
```

should have be

```
17  boolean found = false;
```

c) Syntactical error that occurs once line 19 is reached:

```
7   int i;
```

should have been

```
7   int i = 0;
```

d) Semantical error:

```
18  while (i < s.length() && found) {
```

should have been

```
18  while (i < s.length() && !found) {
```

e) Semantical Error:

```
26  System.out.println("The first occurrence of '" + c +  
27    "' in \"" + s + "\" is a position " + i);
```

should have been:

```
26  System.out.println("The first occurrence of '" + c +  
27    "' in \"" + s + "\" is in position " + i-1);
```

Exercise 5

(8+8+6=22 Marks)

In a computer, all information (including program instructions) is represented as numbers in the binary number system. In this system, there are only two digits, 0 and 1, which are often referred to as bits. A binary editor allows one to view information stored in a computer as series of bits. However, the binary number system is often impractical. Therefore, many binary editors convert binary numbers to the hexadecimal number system before displaying them. The hexadecimal number system has 16 digits: the well-known decimal digits from 0 to 9, and the letters from A to F. Converting a binary number to the hexadecimal number system is fairly straightforward:

- Break down the binary number into groups of 4 bits
- Convert each group of 4 bits to one hexadecimal digit according to the following conversion table:

Binary	Hexadecimal	Binary	Hexadecimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

- a) Implement an iterative method that takes as input a string that represents a binary number and convert the binary number entered by the user to the hexadecimal number system, and returns the resulting hexadecimal number.

Solution:

```
public static String binaryToHexa(String s) {

    String r = "";
    int n1, n2, n3, n4;
    for(int i =0; i<s.length()-1; i=i+4){
        n1 = Integer.parseInt(s.charAt(i)+"");
        n2 = Integer.parseInt(s.charAt(i+1)+"");
        n3 = Integer.parseInt(s.charAt(i+2)+"");
        n4 = Integer.parseInt(s.charAt(i+3)+"");

        int result = n1*8 + n2*4 + n3*2 + n4;
        switch(result) {
            case 10: r = r + 'A'; break;
            case 11: r = r + 'B'; break;
            case 12: r = r + 'C'; break;
            case 13: r = r + 'D'; break;
            case 14: r = r + 'E'; break;
            case 15: r = r + 'F'; break;
            default: r = r + result;
        }
    }
    return r;
}
```

- b) Implement a recursive method to perform the same task as in part a).

Solution:

```

public static String binaryToHexaRec(String s) {
    if (s.length()==0)
return "";
    else {
        int n1 = Integer.parseInt(s.charAt(0)+"");
        int n2 = Integer.parseInt(s.charAt(1)+"");
        int n3 = Integer.parseInt(s.charAt(2)+"");
        int n4 = Integer.parseInt(s.charAt(3)+"");
        int result = n1*8 + n2*4 + n3*2 + n4;
        if (result == 10)
            return 'A' + binaryToHexaRec(s.substring(4,s.length()));
        else if (result==11)
            return 'B' + binaryToHexaRec(s.substring(4,s.length()));
        else if (result==12)
            return 'C' + binaryToHexaRec(s.substring(4,s.length()));
        else if (result==13)
            return 'D' + binaryToHexaRec(s.substring(4,s.length()));
        else if (result==14)
            return 'E' + binaryToHexaRec(s.substring(4,s.length()));
        else
            return 'F' + binaryToHexaRec(s.substring(4,s.length()));
    }
}
}

```

c) Write a class called `ConvertToHexadecimal`, which contains only a `main()` method that does the following:

- Ask the user to enter a String; this String will represent a binary number.
- Convert the binary number entered by the user to the hexadecimal number system, and display the resulting hexadecimal number. You can use one of the methods from the previous parts.
- Ask whether the user wants to convert another binary number to the hexadecimal number system. If the user answers "yes", then the program will repeat the steps listed above, otherwise it will terminate.

Sample session:

```

Enter a binary number: 11011110101011011011111011101111
The corresponding value in hexadecimal: DEADBEEF
Would you like to convert more?
Enter "yes" to continue: yes
Enter a binary number: 11001010111111101011101010111110
The corresponding value in hexadecimal: CAFEBABE
Would you like to convert more?
Enter "yes" to continue: no

```

You may assume that the String entered by the user contains no characters other than '0' and '1'. You may also assume that the length of the String entered by the user is a multiple of 4.

Note: Use a do-while loop to implement the main method.

```

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

```

Solution:

```

public static void main (String[] args) throws IOException {

```

```
InputStreamReader instream= new InputStreamReader(System.in);
BufferedReader stdin= new BufferedReader(instream);
String s;
String a;

do {
System.out.println("Enter a binary number:");
s = stdin.readLine();
System.out.println("The corresponding value in hexadecimal: ");
System.out.println(binaryToHexa(s));
System.out.println("Would you like to convert more?");
System.out.println("Enter \"yes\" to continue:");
a = stdin.readLine();
}
while (a.equals("yes"));
}
```

Extra Sheet

Extra Sheet

Extra Sheet

Extra Sheet