Midterm Examination SOLUTION KEY

This is a closed-book-closed-notes exam
You may use a crib sheet no more than 2 pages long.

Please check that you have all 7 pages.
(the last page is blank)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem 1 (20)</td>
<td></td>
</tr>
<tr>
<td>Problem 2 (15)</td>
<td></td>
</tr>
<tr>
<td>Problem 3 (20)</td>
<td></td>
</tr>
<tr>
<td>(3 extra credit)</td>
<td></td>
</tr>
<tr>
<td>Problem 4 (25)</td>
<td></td>
</tr>
</tbody>
</table>

Total (80 points)
Part 1. The basics. (20 points)

(a) Consider the following code segment and show what is printed when it is executed.

```java
int length = 9;
int pieces = 2;
String str = "strong";
if (length/pieces > 4){
    System.out.println (str.charAt(4));
} else{
    System.out.println (str.length());
}
```

prints 6, because length/pieces is a result of integer
division of 9 by 2, i.e. 4

(b) Consider the following code segment and answer the questions below:

```java
int years = 5;
double debt = 120.3;
double credit = 550.5;
boolean reliable = false;
if (years > 3 || credit < 2*debt && debt < 100 ) {
    if (!reliable)
       System.out.println ("One");
    else
       System.out.println (years%3);
} else {
    if (reliable || credit - debt < 50)
       System.out.println ("Two");
    else
       System.out.println (years/2.0);
}
```

What will be printed? Show the values of the expressions involved for partial
credit.

prints "one", since

years > 3 evaluates to true and so does !reliable
(c) Consider the following code segment and show what is printed when it is executed. Show the intermediate values of variables k and s for partial credit.

```java
int k = 1;
int s = 0;
while (k < 6) {
    System.out.print('*');
    if (k % 2 == 1) // i.e. k is odd
        s = s + k;
    k++;
}
System.out.println(k);
System.out.println(s);
```

The loop body is executed 5 times, and stops executing when k is 6. 

s is incremented by k, when k is odd, i.e. when it is 1, 3 and 5, i.e. s equals 9 after the loop. Printout looks as follows:

```
*****6
9
```

(d) What is printed when the loop below is executed?

```java
String str = "network";
String newstring = "";
int i;
for (i = 0; i < str.length(); i++){
    newstring = '~' + str.charAt(i);
}
System.out.println (newstring);
```

~k
Part 2. (15 points) Static methods.

What is displayed when the following code segment is executed? Someone has created this very poorly written undocumented program. Do not try to deduce the purpose of this code. Simply show the output in the box on the bottom of the page. Show the values of participating variables, parameters and expressions for partial credit.

```java
public class Example {
    public static void main (String[] args ){
        String word = "Kilimanjaro";
        int len = word.length();
        int num = 3;
        char ch = '\#';

        System.out.println ("len=");
        System.out.println ("num=");
        System.out.println ("ch=");

        ch = myMethod(word, num);

        System.out.println ("len=");
        System.out.println ("num=");
        System.out.println ("ch=");
    }

    public static char myMethod( String str, int pos){
        int len = str.length() / 2;
        char aChar = '\$';

        System.out.println ("str=");
        System.out.println ("pos=");
        System.out.println ("len=");
        System.out.println ("aChar=");

        if (pos > len){
            aChar = str.charAt(len+pos);
        } else {
            aChar = str.charAt(len-pos);
        }

        System.out.println ("len=");
        System.out.println ("aChar=");
        return (aChar);
    }
}
```

len = 11, num = 3, ch = #
str = Kilimanjaro, pos = 3
len = 5, aChar = $
len = 5, aChar = 1
len = 11, num = 3, ch = 1
**Part 3 (20 points)** For this problem you need to develop a static method that calculates a discounted price based on a color code value. The method will be passed 2 parameters:

- the original price of a product in dollars,
- a word designating a color code based on which the discount is to be given.

The method should compute and return the discounted price, according to the following rules:

- In case color code is `red`, the original price is discounted by 30%
- In case color code is `blue`, the original price is discounted by 20%
- In case color code is `green`, the original price is discounted by 10%
- For any other value of the color code – no discount is given, and the returned value should be equal to the original price.

For example, when passed values 1000.0, “red” the method should return 700.0, which is the original price discounted by 30%; if passed 1000.00, “blue” it should return 800.0, if passed 1000.00, “green” it should return 900.0. If anything else but “red”; “green” or “blue” is passed as the color code – the method should return the original price.

For extra credit (3 points) make the method insensitive to the case of the color code word, i.e. treat “rEd”, for example, in the same way as “red”.

The template for the method is provided below.

```java
public static double discountPrice ( double oriPrice, String colorCode) {
    double discounted = oriPrice;
    if (colorCode.equalsIgnoreCase("red")){
        discounted = .7 * oriPrice;
    } else if (colorCode.equalsIgnoreCase("blue")){
        discounted = .8 * oriPrice;
    } else if (colorCode.equalsIgnoreCase("green")){
        discounted = .9 * oriPrice;
    }
    return (discounted);
}
```
Part 4. (25 points) Write a complete program that displays discounted prices for a list of products.

The program must read a sequence of price values, and color code strings and must print the discounted price according to the color code as described in Part 3. The program must terminate when the user enters 0 for the original price value, printing out a total number of products that had color code equal to red.

The following sample interactions demonstrate the desired behavior of the program: User input is indicated in boldface.

Please enter the original price: 10.10
Please enter the color code: red
The discounted price is $7.07

Please enter the original price: 20.0
Please enter the color code: green
The discounted price is $16.0

Please enter the original price: 100.0
Please enter the color code: red
The discounted price is $70.0

Please enter the original price: 150.0
Please enter the color code: blue
The discounted price is $120.0

Please enter the original price: 0
TOTAL NUMBER OF RED: 2

To get full credit, your program must use method discountPrice from Part 3 (you do not have to repeat the definition of that method –just show where that definition should go). A solution that does not use this function will get at most 20 points.

Your program doesn’t have to contain comments, but must otherwise use good programming style. Partial credit will be awarded.
public class Discount {
    public static void main (String [] args) {
        double price, discPrice;
        String color;
        int numRed = 0;

        System.out.println("Please enter the original price:");
        price = SavitchIn.readLineDouble();

        while (price > 0) {
            System.out.println("Please enter the color code:");
            color = SavitchIn.readLine();

            discPrice = discountPrice (price, color);
            System.out.println("The discounted price is \$"+discPrice);
            if (color.equalsIgnoreCase("red"))
                numRed ++;

        }

        System.out.println("TOTAL NUMBER OF RED:" + numRed);
    }

    // discountPrice() method goes here..
}