Midterm Examination

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

<table>
<thead>
<tr>
<th>Part 1 (20 points)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 2 (20 points)</td>
<td></td>
</tr>
<tr>
<td>Part 3 (15 points)</td>
<td></td>
</tr>
<tr>
<td>Part 4 (30 points)</td>
<td></td>
</tr>
</tbody>
</table>

Total (85 points)

Read these instructions first:

1. When starting to work on a problem – **read the problem carefully and make sure you understand what you’re asked to do**. You should ask the instructor to clarify anything that is unclear in the problem statement.

2. **Always show your work** – that is the only basis for partial credit.

3. **Do not blindly copy from your crib sheet – THINK!** Valid ideas count for more than blind copying – as they are your own!

4. **Always check your answers** and test your code on a sample input.

5. Minor syntax errors such as, for instance, missing ; in your code will not be counted as an error.

Good luck!
**Part 1. (20 points) The basics.** Show your work for partial credit

(a) Consider the following code segment:

```java
String myStr = SavitchIn.readLine();
int len = myStr.length();
String result;

if (len <= 3 || myStr.charAt(0) == 'Z') {
    result = "RARE";
} else {
    result = "NOT RARE";
}
```

Show values of variables len and result after the execution of this program segment for the following user inputs:

- **user enters** Zanzibar  
  - len: 8  
  - result: RARE

- **user enters** Moo  
  - len: 3  
  - result: RARE

(b) What will be printed by the following code segment? PRINTER Two

```java
int a = 20;
int b = 100;
boolean flag = false;

if (b % a == 0)
    flag = true;

if (!flag) {
    System.out.println("One");
}
else {
    System.out.println("Two");
}
```

Show values of following expressions:

- b % a: 0
- !flag: false
(c) Write down a code segment that uses a \texttt{while} loop that produces exactly the same output as the following \texttt{for} loop.

```java
for (int i = 12; i <= 24; i = i + 2) {
    System.out.print (i / 3);
}
```

```java
int i = 12;
while (i <= 24) {
    System.out.print (i / 3);
    i = i + 2;
}
```

(d) What is printed when the following code segment is executed?

```java
for (int i = 1; i <= 5; i++) {
    sum = 0;
    count = 0;
    while (sum < 8) {
        System.out.print (i);
        sum = sum + i;
        count ++;
    }
    System.out.println ("count is:" + count);
}
```

```
1 1 1 1 1 1 1 count is:8
2 2 2 2 count is:4
3 3 3 count is:3
4 4 count is:2
5 5 count is:2
```
Part 2. (20 points) 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 ){
        int hour = 14;
        int m = 28;
        System.out.print ("hour = " + hour + " minute = " + m);
        hour = myMethod(hour, m);
        System.out.print ("hour = " + hour + " minute = " + m);
    }

    public static int myMethod( int h, int m){
        int hour;
        if (h < 8 ) {
            h = 8;
            m = 0;
        }
        else if ( m < 30 && h > 15 || m > 25 ) {
            h = 12;
            m = 55;
        }
        else {
            h = m / h;
            m = m % h;
        }
        System.out.print ("h is " + h + " m is " + m);
        hour = m + 10;
        return h;
    }
}
```

```
hour = 14 minute =28
h is 12 m is 55
hour = 12 minute =28
```
Part 3. (15 points) In this problem you will develop a static method `bankAccountUpdate` that calculates an updated balance value. The method will be passed 3 parameters:

- The current balance in dollars.
- A character representing the transaction type: ‘w’ or ‘W’ for withdrawal, ‘d’ or ‘D’ for deposit. Any other character should be treated as invalid transaction code, in which case the balance should stay unchanged.
- The dollar amount of the transaction.

The method should compute and return the updated balance value that is equal to

- the original balance + amount, for the deposit
- the original balance - amount, for the withdrawal
- the original balance, for an invalid transaction.

For example, when passed values 1100.05, ‘D’, 300.00 the method should return 1400.05 which is equal to 1100.05 + 300. If passed 500.00, ‘w’, 600.00, it should return -100.00.

The template for the method is provided below.

```java
public static double bankAccountUpdate(double balance, char tType, double amount) {
    double newBalance;
    switch (tType) {
        case 'w':
            case 'W':
                newBalance = balance + amount;
                break;
        case 'd':
            case 'D':
                newBalance = balance - amount;
                break;
        default:
            newBalance = balance;
    }
    return (newBalance);
}
```
Part 4. (30 points) Write a complete program that lets the user balance a checkbook.

The program must read the the value of the original balance, followed by a sequence of transaction type code (‘w’ or ‘W’ for a withdrawal, ‘d’ or ‘D’ for a deposit) and transaction amount pairs. The program must terminate when either the balance becomes negative or the user enters character ‘S’ (for Stop) instead of a valid transaction code.

The program should output
1. the final balance,
2. the total amount of money deposited,
3. the total amount of money withdrawn.

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

Please enter the original balance: 300.25
Please enter transaction type: W
Please enter amount: 20.25
Your updated balance is $280.0

Please enter transaction type: d
Please enter amount: 50.35
Your updated balance is $330.35

Please enter transaction type: W
Please enter amount: 10
Your updated balance is $320.35.

Please enter transaction type: S

PRINTING FINAL VALUES:
BALANCE: 320.35
TOTAL DEPOSITED 50.35
TOTAL WITHDRAWN 30.25

Here’s another example in which the program terminated when the balance dropped below zero.

Please enter the original balance: 300.25
Please enter transaction type: D
Please enter amount: 200.25
Your updated balance is $500.50

Please enter transaction type: d
Please enter amount: 50.35
Your updated balance is $550.85

Please enter transaction type: W
Please enter amount: 600
Your updated balance is $-49.15

PRINTING FINAL VALUES:
BALANCE: -49.15
TOTAL DEPOSITED 250.60
TOTAL WITHDRAWN 600.0

To get full credit, your program must use the method bankAccountUpdate from Part 3. A solution that does not use this function will get at most 25 points.

Your program doesn’t have to contain comments, but must otherwise use good programming style. Partial credit will be awarded.
public class Balance {
    public static void main (String[] args) {

        double balance; // current balance
        double totalW=0, totalD=0; // total withdrawn & deposited
        double amt; // amount of transaction
        char trans; // type of transaction

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

        System.out.println("Please enter transaction type:")
        trans = SavitchIn.readLineNonwhiteChar();

        while (trans != 'S' && balance >= 0) {

            System.out.println("Please enter amount: ")
            amt = SavitchIn.readLineDouble();

            balance = bankAccountUpdate(balance, trans, amt);
            System.out.println("Your updated amount is: "+ balance);

            if (trans == 'w' || trans == 'W')
                totalW = totalW + amt;
            else if (trans == 'd' || trans == 'D')
                totalD = totalD + amt;

        }

        if (balance >= 0){
            System.out.println("Please enter transaction type:")
            trans = SavitchIn.readLineNonwhiteChar();
        }

        System.out.println("PRINTING FINAL VALUES:");
        System.out.println("BALANCE:" + balance);
        System.out.println("TOTAL DEPOSITED": + totalD);
        System.out.println("TOTAL WITHDRAWN": + totalW);

    */
    /* definition of bankAccountUpdate method from part 3 goes here */
}