

Introduction to Computer Programming

Spring term 2009

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 11 pages (6 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	6	Σ
Possible Marks	6	8	8	13	10	10	55
Final Marks							

Exercise 1

(6 Marks)

Implement a method `discount` which takes two parameters. The first parameter represents the sub-total before any discount is applied. The second parameter represents the customer rating. The method calculates and returns the final amount due after the customer discount is applied as follows:

- If the customer rating is 1, there is 10% discount.
- If the customer rating is 2, there is 25% discount.
- If the customer rating is 3, there is 50% discount.
- Any other ratings receives no discount.

The method should use the switch statement.

Do not write a main method.

Solution:

```
public static double discount(double subTotal, int rating) {
    switch (rating) {
        case 1:
            return subTotal * 0.90;
        case 2:
            return subTotal * 0.75;
        case 3:
            return subTotal * 0.5;
        default:
            return subTotal;
    }
}
```

Exercise 2

(8 Marks)

Implement a method that takes an integer as an argument and returns a string representing the binary representation of the integer. Given the argument 42, it should return 101010. Given the argument -42, it should return -101010.

Note: Use one single do-while loop to solve the problem.

Do not write a main method.

Solution:

```
public static String i2s(int x) {
    String sign = "";
    String binary = "";
    if (x < 0)
        {sign = "~"; x *= -1;}
    do {
        int low = x % 2;
        binary = low + binary;
        x /= 2;
    } while (x > 0);

    return sign + binary;
}
```

Exercise 3

(4+4=8 Marks)

- a) What is the output of the following fragment of code? Trace your code!

```
int N = 4;
for (int i = 0; i < N; i++)
    for (int j = 0; j < N; j++)
        if (i != j) System.out.println(i + ", " + j);
```

Solution:

```
0, 1
0, 2
0, 3
1, 0
1, 2
1, 3
2, 0
2, 1
2, 3
3, 0
3, 1
3, 2
```

- b) What is the output of the following fragment of code? Trace your code?

```
int N = 4;
for (int i = 0; i < N; i++)
    for (int j = 0; (i != j) && (j < N); j++)
        System.out.println(i + ", " + j);
```

Solution:

```
1, 0
2, 0
2, 1
3, 0
3, 1
3, 2
```

Exercise 4

(8+5=13 Marks)

- a) Write a Java method `triangle` that takes two integers `seed` and `level` and displays a pattern like the one given in the examples below on the screen.

For example

- `triangle(3,4)`, where 3 is the `seed` value and 4 is the `level` value, will display

```
3
4 5
6 7 8
9 0 1 2
```

- `triangle(7,5)`, where 7 is the `seed` value and 5 is the `level` value, will display

```
7
8 9
0 1 2
3 4 5 6
7 8 9 0 1
```

- b) Write a main method that reads from the user the values of `seed` and `level` and calls the method above.

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

Solution:

```
import java.io.*;

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

        String inData;
        System.out.println("Enter the seed value");
        inData = stdin.readLine();
        int seed = Integer.parseInt(inData);
        System.out.println("Enter the size");
        inData = stdin.readLine();
        int size = Integer.parseInt(inData);
        drawPattern(seed, size);
    }

    public static void drawPattern(int seed, int size)
    {
        int i, j;
        for(i = 1; i<=size; i++) {
            for(j = 1; j<=i; j++) {
                if(seed>9)
                    seed = 0; //reinitialize
```

```
        System.out.print(seed + " ");
        seed++;
    }
    System.out.println();
}
}
```



Exercise 5

(10 Marks)

Write a recursive method `makeUnits` that takes an integer parameter representing the number of cents and prints out the number of quarters, dimes, nickels, and cents that can be made out of this amount. quarter = 0.25 dollar, dime = 0.1 dollar, nickel = 0.05 dollar, cent = 0.01 dollar.

For example `makeUnits(92)` prints out the following

The amount is

2 Cent

1 Nickel

1 Dime

3 Quarter

Do not write a main method.

Solution:

```
public static void makeUnits(int cents)
{
    if(cents<5)
    {
        System.out.println("The amount is:");
        System.out.println(cents+" Cents");
        return;
    }
    if(cents<10)
    {
        makeUnits(cents%5);
        System.out.println(cents/5+" Nickels");
    }
    else if(cents<25)
    {
        makeUnits(cents%10);
        System.out.println(cents/10+" Dimes");
    }
    else
    {
        makeUnits(cents%25);
        System.out.println(cents/25+" Quarters");
    }
}
```

Exercise 6

(10 Marks)

Consider the following recursive method:

```
public static void mystery(String prefix, String remaining, int k) {
    if (k == 0) {
        System.out.println(prefix);
        return;
    }
    if (remaining.length() == 0) return;
    mystery(prefix + remaining.charAt(0), remaining.substring(1), k-1);
    mystery(prefix, remaining.substring(1), k);
}
```

- a) What value is returned by the following invocation? Trace your program!

```
mystery("", "CSEN", 3)
```

Solution:

```
CSE
CSN
CEN
SEN
```

- b) Give a concise verbal description of how the value returned by `mystery("", s, k)` is related to its parameter values `s` and `k`.

Solution:

The method call `mystery("", s, k)` prints out all subsequences of `s` of length `k`.

Extra Sheet

Extra Sheet

Extra Sheet

Extra Sheet