### AYBUKE BUYUKCAYLI – KORAY OZUYAR – MUSTAFA SOYLU

Week 21/02/2007-23/02/2007 Lecture Notes:



### Example java program

Draw a flowchart to count the number of even and odd numbers. Stop if input no. <= 0. (Assume input no. >=0)

n= input no.

noEven= number of even numbers

noOdd= number of odd numbers

continue= means that input is ok. and continue with counting even and odd numbers.(boolean value) Here is the flowchart:



Pseudo code: (Here is the pseudo code type of the previous program that counts odd and even numbers)

```
int noEven, noOdd;
boolean continue, even;
noEven=0;
noOdd=0;
read n
continue=(n>0)
while (continue) {
      even = (n\%2 = = 0)
      if (even)
        noEven++
      else
        noOdd++
      end if
      read n
      continue=(n>0)
{
    print noEven, noOdd
    stop
```

3

```
import java.util.Scanner;
public class CountEvenOdd
{
    public static void main( String [] args)
    {
        int noEven = 0,n;
        int noOdd = 0;
        boolean continue,even;
    }
}
```

```
Scanner scan = new Scanner(System.in); // We make a new object.
```

System.out.print( "Enter an integer, to stop you may enter 0 or a negative number:");

```
n = Scan.nextInt();
cont = (n > 0);
while(continue)
{
    even = (n % 2 = = 0);
    if(even)
        noEven + +; // (noEven = noEven + 1)
    else
        noOdd + +;
    System.out.print("Enter next integer:");
    n = Scan.nextInt();
    count = (n > 0);
```

```
} //end of while
System.out.println("noOfEven:" + noEven + "noOfOdd" + noOdd);
```

- } //end of main method
- } //end of CountEvenOdd (end of class)





# Example:

x = a + (b - c) / d;



Example:

$$a = ((b + (c^* d)) - ((e / f) % x));$$

a = b + c \* d - e / f % x;



| 17 % 4   | 1   | sign is the sign of the numerator |
|----------|-----|-----------------------------------|
| - 20 % 3 | -2  |                                   |
| 20 % -3  | 2   |                                   |
| 10 % -5  | 0   |                                   |
| 3 % 8    | 3   |                                   |
| 10 / 4   | 2   |                                   |
| 10 / 4,0 | 2,5 | ( integer / double)               |
| 10,0 / 4 | 2,5 | ( doble / integer)                |

© If one of the number is a floating number, the result should be a floating number.

# INCREMENT AND DECREMENT OPERATORS

count ++; (post increment)

count--;(post decrement)

++ count; (pre increment)

Here are some examples to help you to understand clearly:



Don't forget that we first assign and then increment the value!!!

| <mark>2)</mark> count = 15; | count | total |
|-----------------------------|-------|-------|
| total = 0:                  | 15    | 0     |
|                             | 16    | 16    |
| total = ++ count;           |       |       |

#### Assignment operators:

total = total+1; total + = 5;  $\left.\right\}$  (these are the same)

result = result\* 
$$(n1 + n2);$$

result \* = (n1+n2);

#### Data conversion:

In java, conversions can occur in three ways:

- a) assignment conversion
- b) promotion
- c) casting

### 1) Widening

Byte a = 5;

( a < b)

Short b = 10;

a = b; \_\_\_\_\_ compile time error

(because we can not try assign a big value into a short value)

# 2) Promotion





### Java Cordinate System



\* We have two important declerations:

```
1.) import.javax.swing.JApplet;
```

```
2.) import.java.awt.*;
```

```
public class Einstein extends JApplet
{
    public void paint(Graphics page)
    {
        Page.drawRect(50,50,40,40); //square
        Page.drawRect(60,80,225,30);
        Page.drawOval(75,65,20,20);
    }
}
```

## <u>Questions</u>

1) In which case the *postfix* and *prefix* forms of the operators (such as total++, ++total) have same results and in which case they don't have the same functions? (Explain the cases by giving specific examples for each one!)

2) What are the main three techniques of data conversions in Java and what is the purpose of this process?

3) What is the main reason of the equivalence in the result; although two different usages of assignment operators?

4) When we are writing an applet, what declerations should we write in the beginning?

5) In " public void main (String [] args) ",why do we use void,what does it mean?

6) How many bits does Unicode have?

7) What is the order of operator precedence? When the java associates it R to L or L to R?

8) When we use the remainder "%" what is the role of it? Can it changes the sign of the numerator every time?

9) In data conversion can we try to assign a big value in to short value? or opposite of it?

10) What is the type casting in java and give an example?

1) *Prefix* and *postfix* operators have same functions when they aren't written as a statement by itself. For instance;

total++; ++total; both operators have same function.

### However; int sum; int total = 5; sum = total++;(This operator assigns the value of total to 6 and then sum is assigned to 6) sum = ++total;(This operator assigns both the total's and sum's value to 6)

2) The three main data conversions in Java are; assignment conversion, promotion and casting. Through widening and narrowing conversions data can be passed without losing data.

3) All of the assignment operators evaluate the expression on the right hand side first, then use the result as right operand of other operation.

division/= num1-num2; is equal to division= division/(num1-num2);

4) import.javax.swing.JApplet; import.java.awt.\*;

5) We use 'void' because it means that it doesn't return anyhing.

6) Unicode has 16 bits.

# 7)

| highest<br>precedend | Precedence      | Operator   | Associates   |
|----------------------|-----------------|--|--------------|
|                      | <sup>ce</sup> 1 | + } unary  | R to L       |
|                      | 4               | *<br>/<br>%  | L to R       |
|                      | 5               | + } Addition<br>- } Substraction<br>+ } String concatenation | L to R<br>on |
| lowest<br>precedend  | 4<br>ce         | - } Assignment   | R to L       |

8) Remainder in java gives remainder of to two dividing number. Remainder can not changes the sign of the numerator. I changes, If and only if the sign is the numerator is negative.

#### Example:

| - 33 % 5 | -3 |
|----------|----|
| 33 % -5  | 5  |

9) In data conversion we can assign short value to big value, however, we cannot assign big value to small value. If we try to assign big value to small value the program gives compile time error.

**10)** Type casting changes the type of the variable by specifying the type in parentheses. It is placed in front of the value to be converted.

#### Example:

| float Money = 18,53;               | money | dollar |
|------------------------------------|-------|--------|
| int dolar ;<br>dolar = (int) money | 18,53 | 18     |