CSCI1530 Computer Principles and Java Programming

Tutorial 6
Flow chart design & Looping

Zheng Qingqing
SHB 911
qqzheng@cse.cuhk.edu.hk

Content

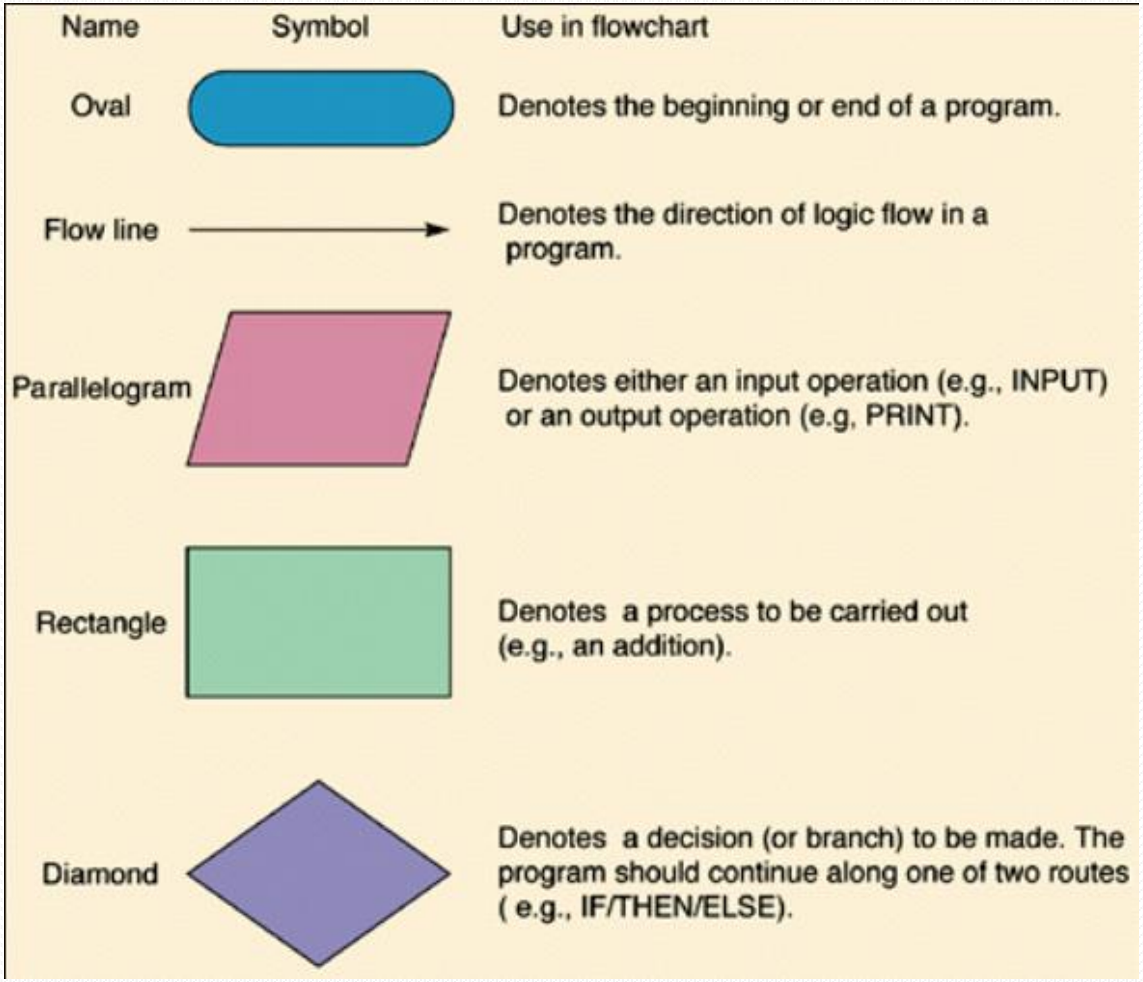
- Flow chart design
  - Flow chart symbol
  - Example 1: GCD
  - Example 2: Print Triangle Stars
- Looping
  - While-loop
  - Nested-while-loop
  - Common errors
- Assignment 3 discussion

Flow chart design

Flow chart

- What: diagram to represent an algorithm, workflow or process.
- How: show the steps as boxes of various kinds, and their order by connecting them with arrows.
- Why: help illustrate a solution model to a given problem; analyze or manage a process in various fields.

Flow chart symbol



Name	Symbol	Use in flowchart
Oval	⬭	Denotes the beginning or end of a program.
Flow line	→	Denotes the direction of logic flow in a program.
Parallelogram	▱	Denotes either an input operation (e.g., INPUT) or an output operation (e.g, PRINT).
Rectangle	▭	Denotes a process to be carried out (e.g., an addition).
Diamond	◇	Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IF/THEN/ELSE).

Example 1: GCD

- Greatest Common Divisor (GCD)
  - Also known as Highest Common Factor (HCF)
    Highest Common Divisor(HCD)

- greatest positive integer that divides the given numbers without a remainder - *from Wikipedia*
  - E.g.: GCD of 18 and 48 is 6.

Workflow of GCD

- Rules
  1. gcd(a, a) = gcd(a, 0) = a
  2. Commutative law:

     gcd(a, b) = gcd(b, a)
  3. assumed a>b, gcd(a, b) = gcd(b, a mod b)
- Idea: repeatedly reduce a & b
- Example: gcd(18, 48)

  = gcd(48, 18)

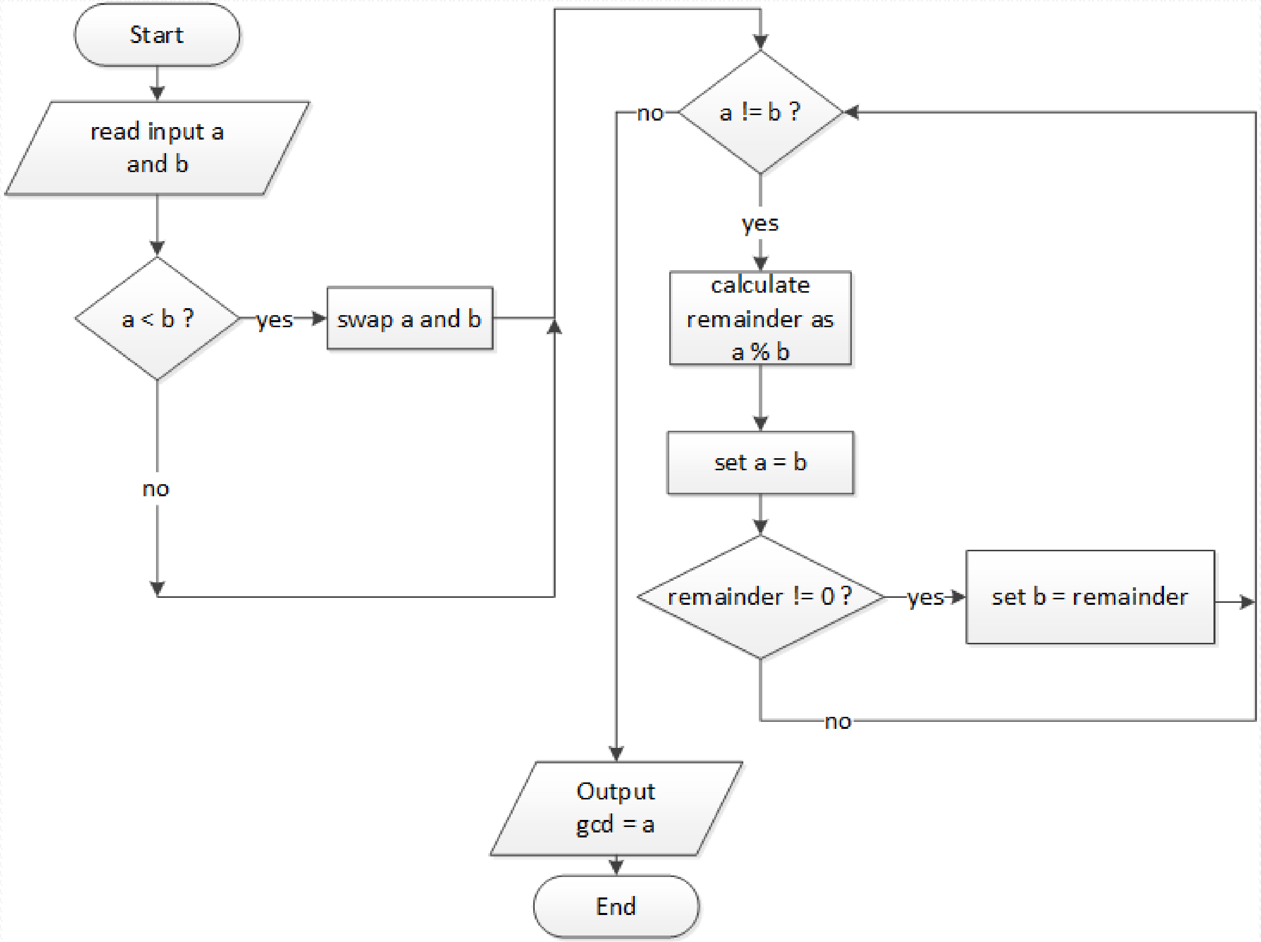
  = gcd(18, 48 % 18) = gcd(18, 12)

  = gcd(12, 18 % 12) = gcd(12, 6)

  = gcd(6, 12 % 6) = gcd(6, 0) = gcd(6, 6)

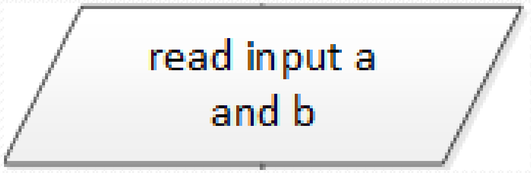
  = 6

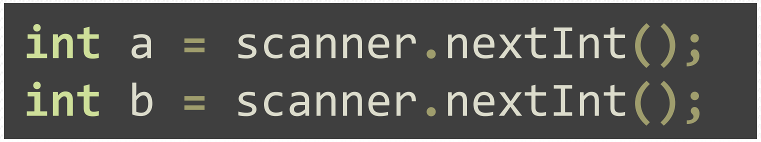
Flow chart of GCD



Building block → Source code

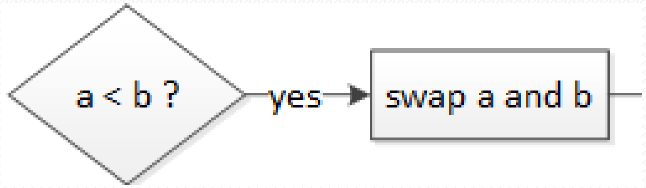
- Input/Output

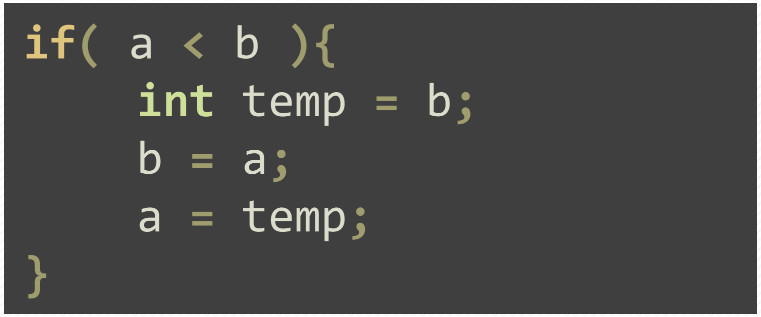




```
int a = scanner.nextInt();
int b = scanner.nextInt();
```

- Conditional or decision

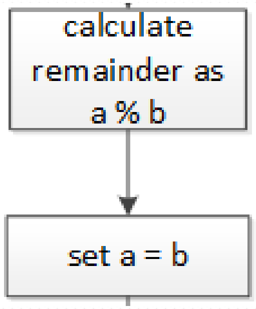


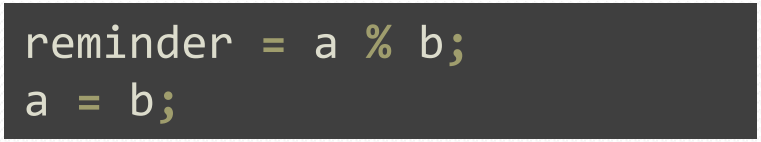


```
if( a < b ){
    int temp = b;
    b = a;
    a = temp;
}
```

Building block → Source code

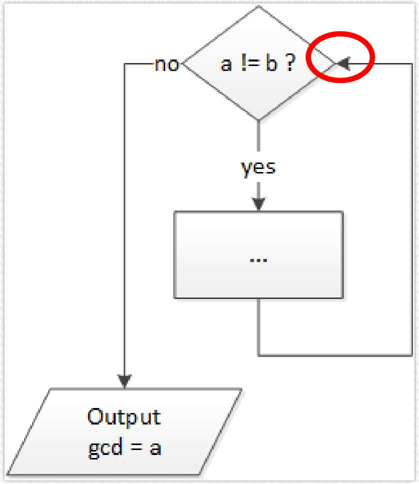
- Generic processing steps

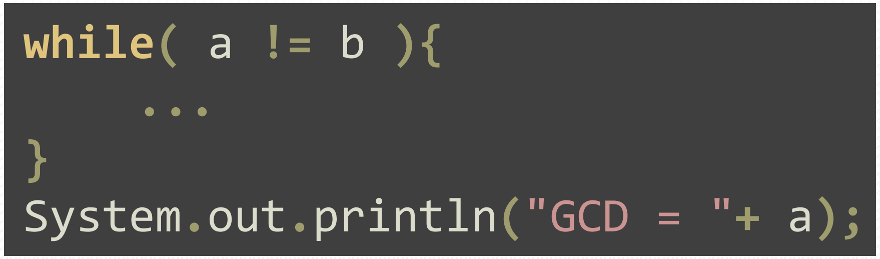




```
reminder = a % b;
a = b;
```

- Junction





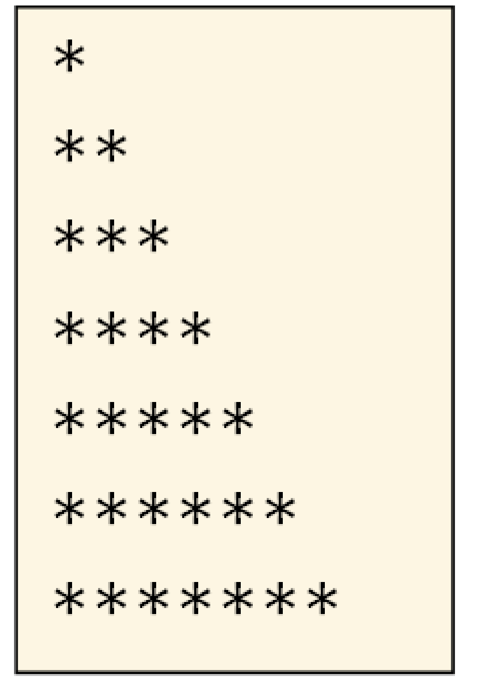
```
while( a != b ){
    ...
}
System.out.println("GCD = "+ a);
```

Example 2: Print triangle stars

- print a triangle of "stars"; the height could be any positive integer n.

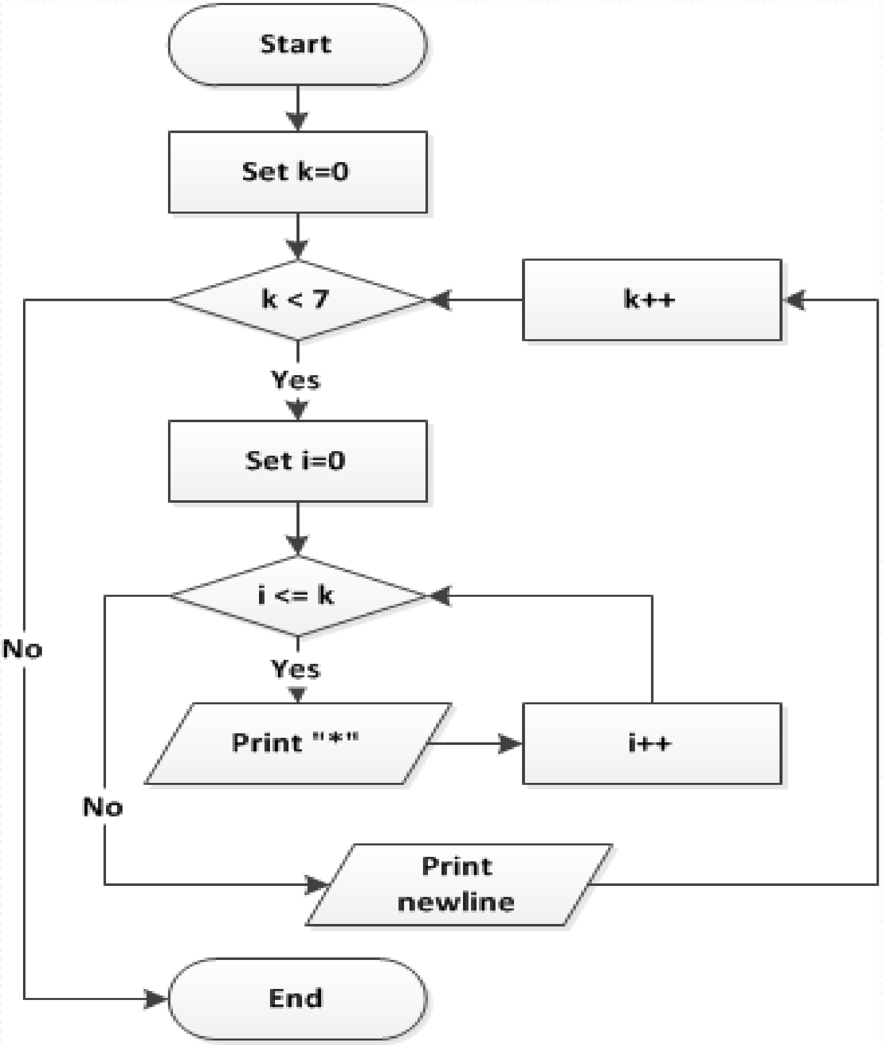
Here n = 7

Output:

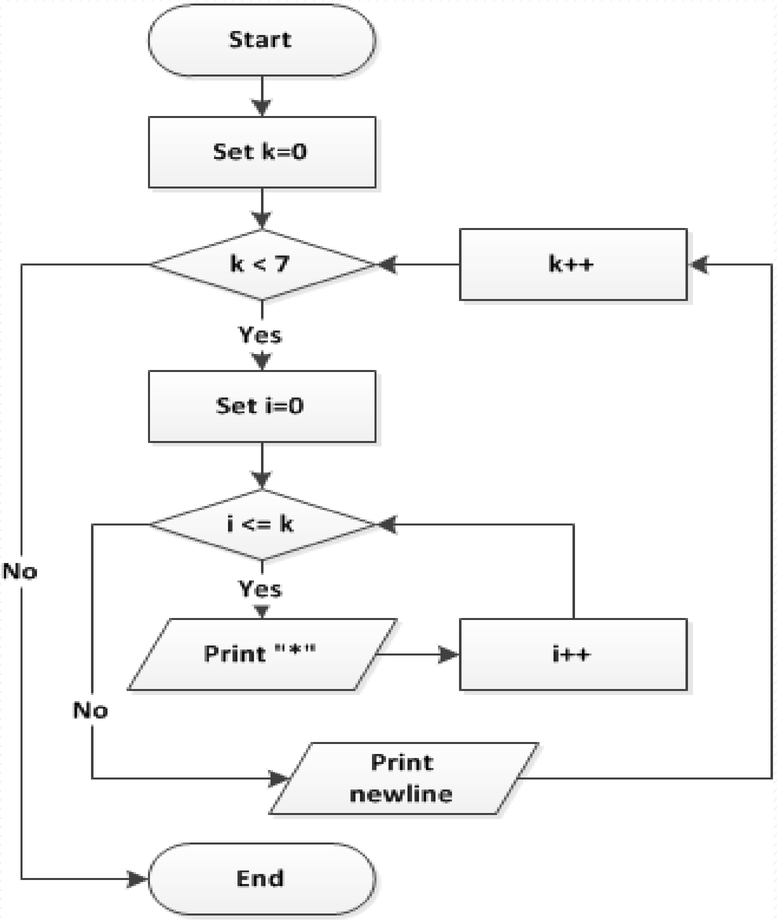


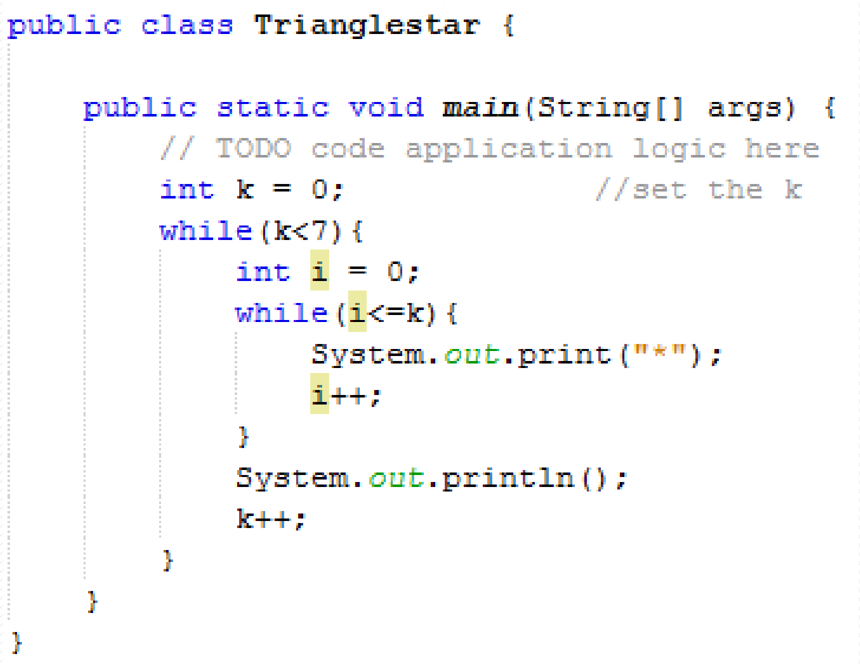
```
*
**
***
****
*****
******
*******
```

Workflow of triangle stars



Flow chart → Source code





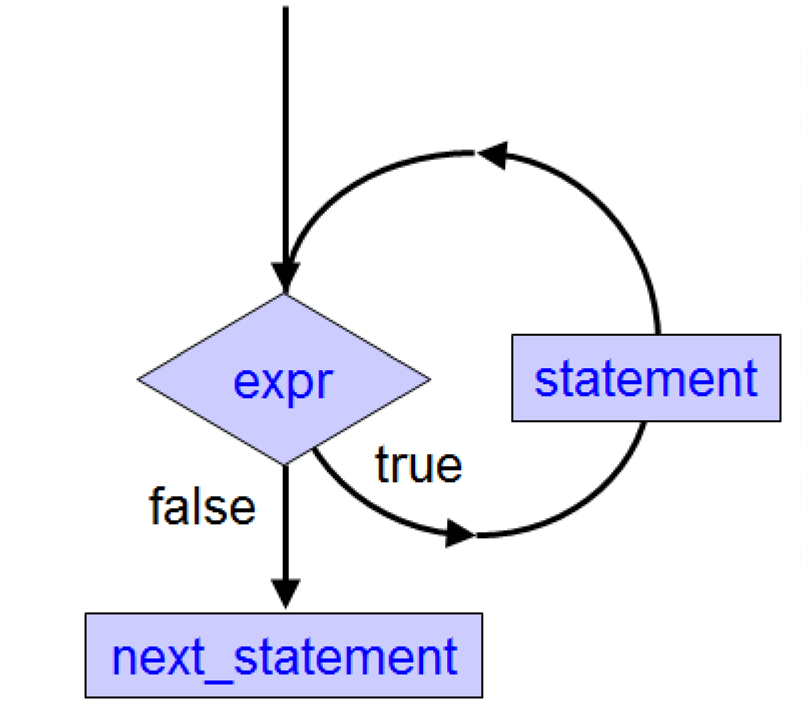
```
public class Trianglestar {

    public static void main(String[] args) {
        // TODO code application logic here
        int k = 0;                  //set the k
        while(k<7){
            int i = 0;
            while(i<=k){
                System.out.print("*");
                i++;
            }
            System.out.println();
            k++;
        }
    }
}
```

Looping

Looping statements

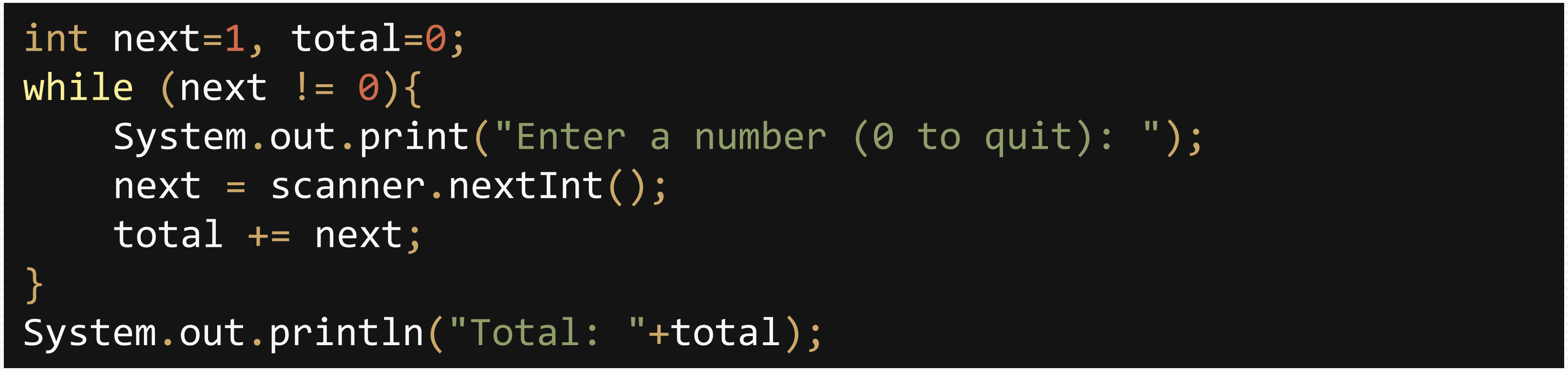
- while-loop
  - Execution of the next iteration depends on an expression

```
while (expr){
    statement;
}
next_statement;
```



Example -- while-loop

- Adding a set of numbers

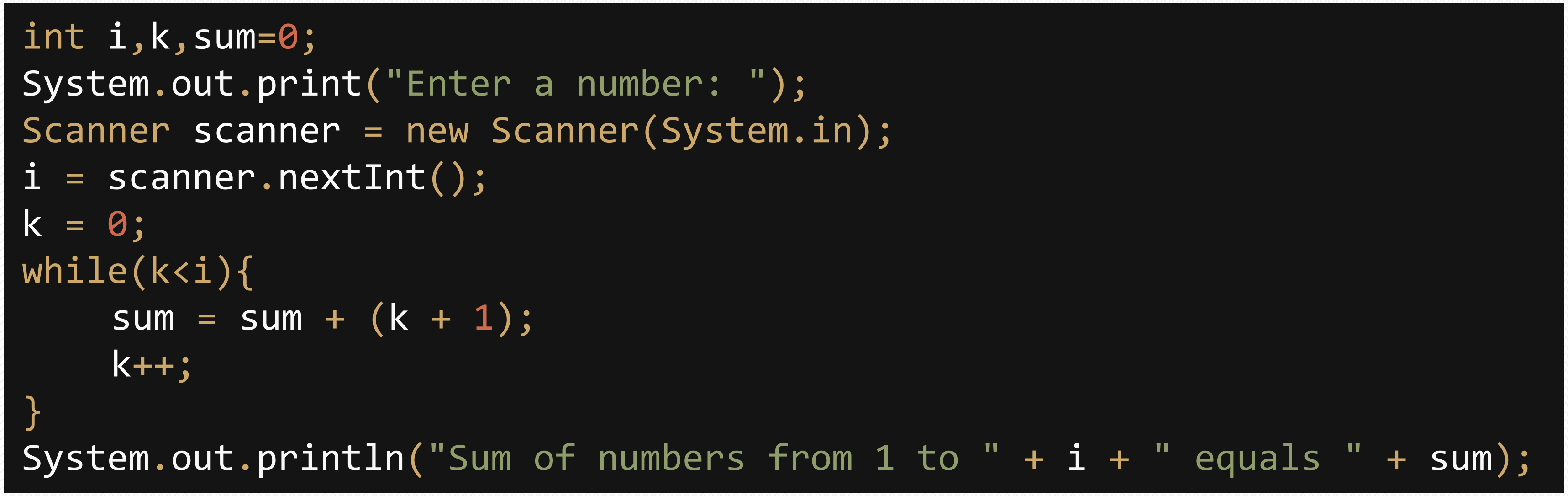


```
int next=1, total=0;
while (next != 0){
    System.out.print("Enter a number (0 to quit): ");
    next = scanner.nextInt();
    total += next;
}
System.out.println("Total: "+total);
```

- Looping depends on an expression
    - "next != 0"

Example -- while-loop

- Sum up numbers start from 1

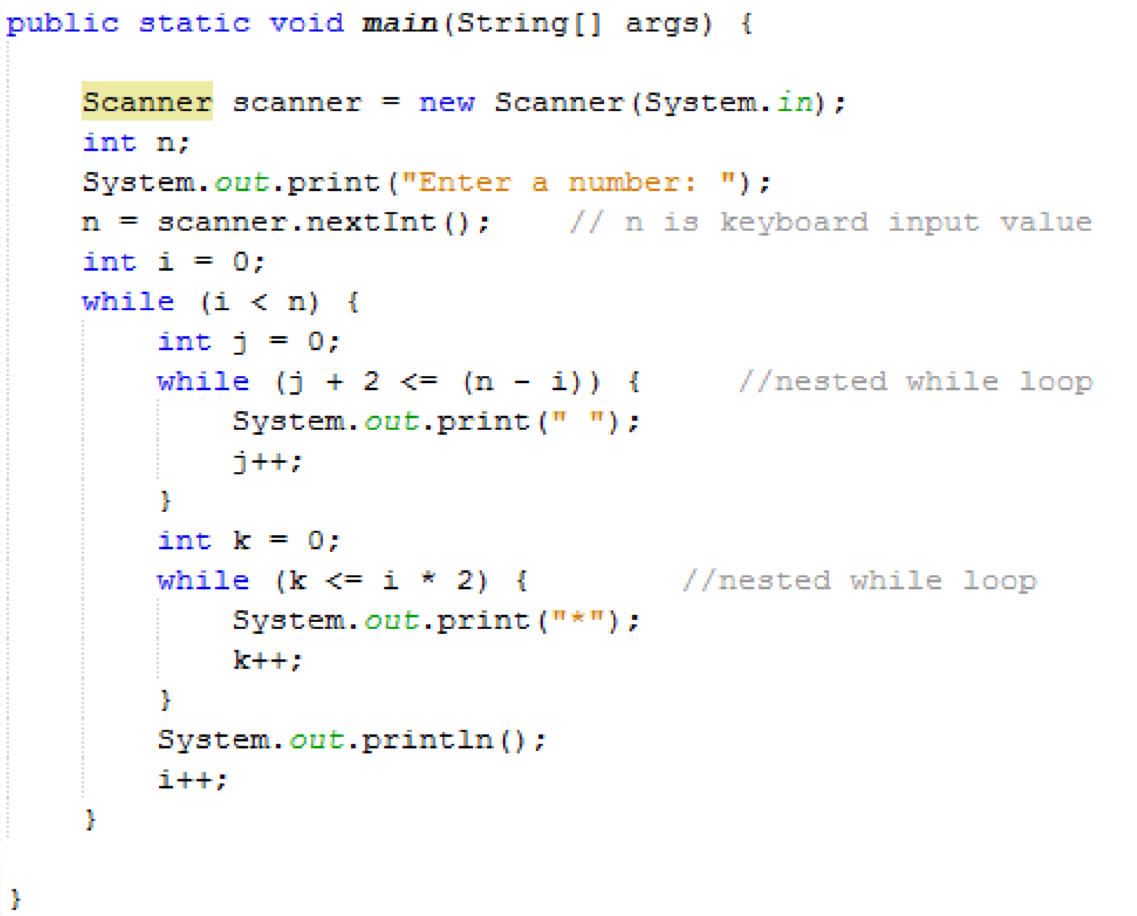


```java
int i,k,sum=0;
System.out.print("Enter a number: ");
Scanner scanner = new Scanner(System.in);
i = scanner.nextInt();
k = 0;
while(k<i){
    sum = sum + (k + 1);
    k++;
}
System.out.println("Sum of numbers from 1 to " + i + " equals " + sum);
```

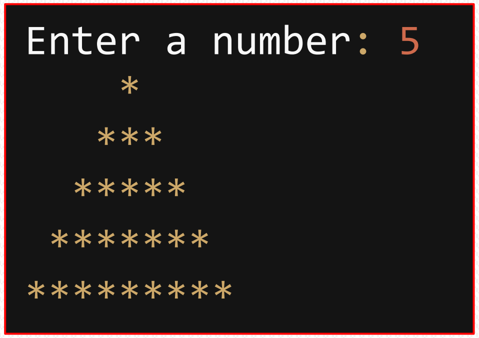
- Use a new variable: k
- Simple way if number of iteration is known
  - Initialize k=0, set k < 'number of iteration'

Example – nested-while-loop

- Nested loop – print a triangle



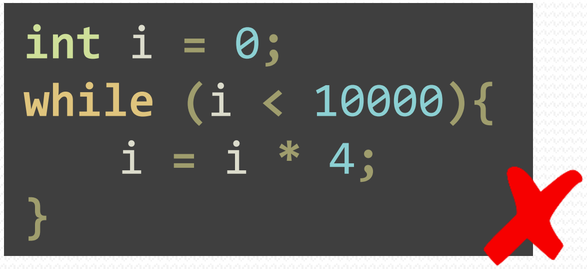
```
public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);
    int n;
    System.out.print("Enter a number: ");
    n = scanner.nextInt();    // n is keyboard input value
    int i = 0;
    while (i < n) {
        int j = 0;
        while (j + 2 <= (n - i)) {     //nested while loop
            System.out.print(" ");
            j++;
        }
        int k = 0;
        while (k <= i * 2) {          //nested while loop
            System.out.print("*");
            k++;
        }
        System.out.println();
        i++;
    }

}
```



```
Enter a number: 5
    *
   ***
  *****
 *******
*********
```

Common errors

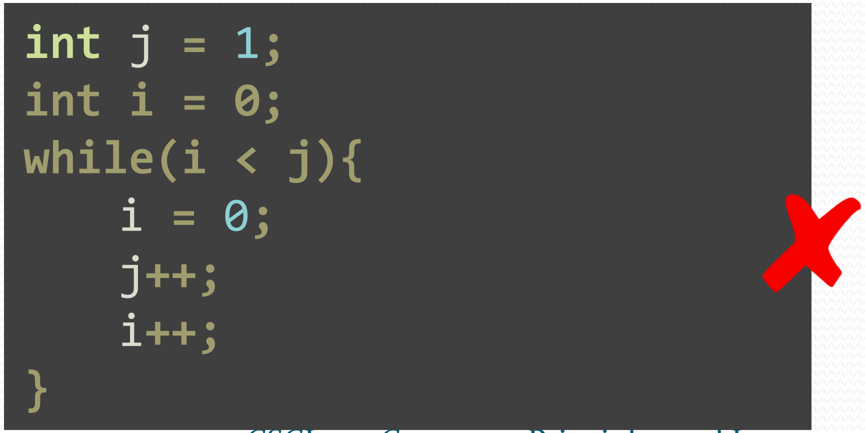
- Infinite loop
  - Initialize with wrong value



```
int i = 0;
while (i < 10000){
    i = i * 4;
}
```

✗

  - Values keep increasing/decreasing in the condition

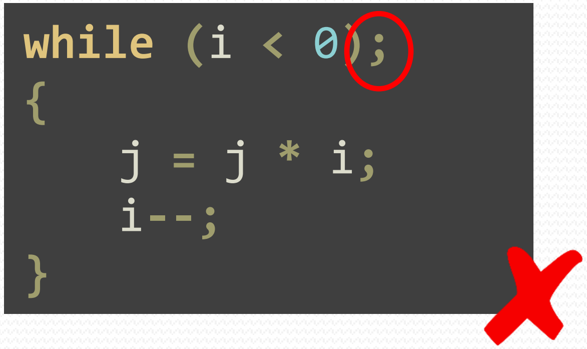


```
int j = 1;
int i = 0;
while(i < j){
    i = 0;
    j++;
    i++;
}
```

✗

Common errors

- Semi-colon ';' at wrong place

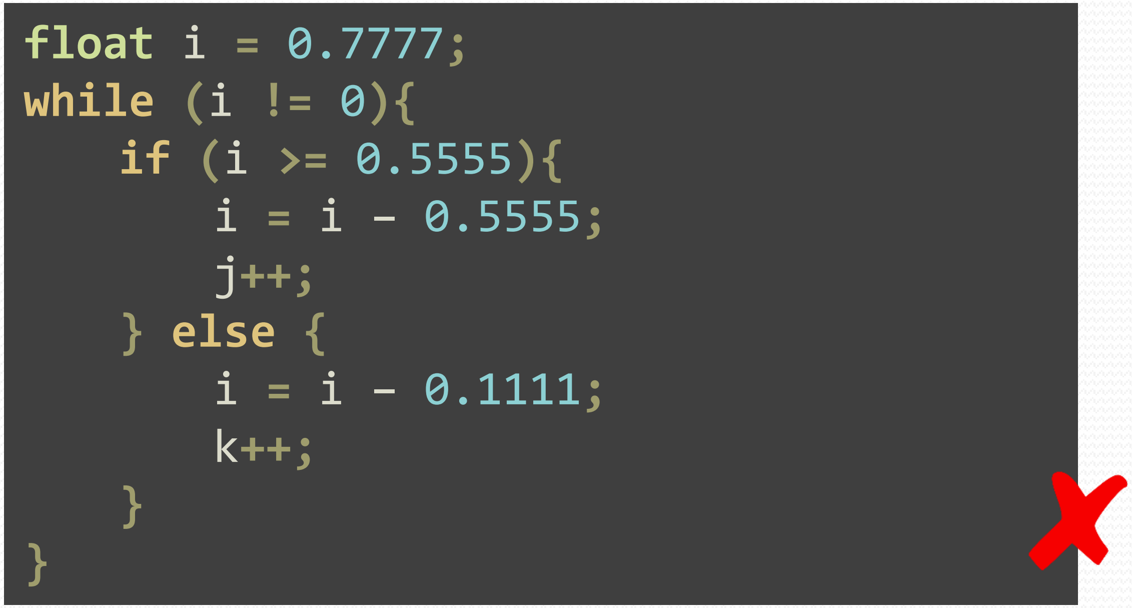


```
while (i < 0);
{
    j = j * i;
    i--;
}
```

  - Nothing has been executed inside the loop
  - 'i' will not be changed
  - Infinite loop

Common errors

- Using equality operators (==, !=) on comparing floating-point number



```
float i = 0.7777;
while (i != 0){
    if (i >= 0.5555){
        i = i - 0.5555;
        j++;
    } else {
        i = i - 0.1111;
        k++;
    }
}
```

✗

Common errors

- Using equality operators (==, !=) on comparing floating-point number
  - Floating-point numbers may have small difference to zero after computations
  - →Infinite loop
  - Better to use integer to count
  - Or, use relational operators (>, >=, <, <=)

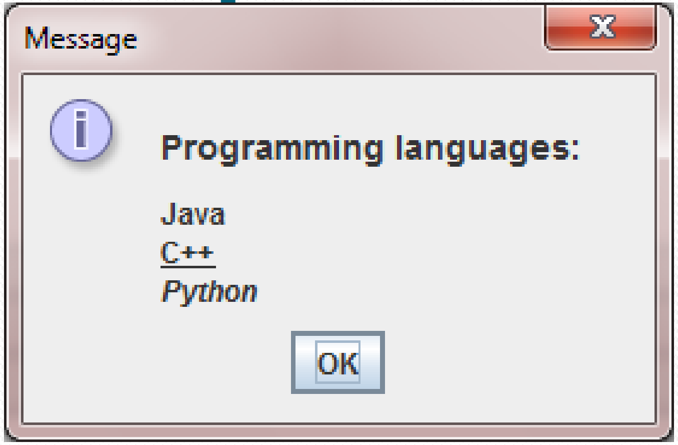
Assignment 3

Hint

- The number you guess:
  - out of guessing bound
  - in the guessing bound:
    - equal to the target
    - smaller than the target:
      - new range [guess+1, range end]
    - larger than the object target:
      - new range[range start, guess-1]

JOptionPane -- HTML Output

```
JOptionPane.showMessageDialog(null,
    "<html>" +
    "<h3>Programming languages:" +
    "<p>Java</p>" +
    "<p><u>C++</u></p>" +
    "<p><i>Python</i></p>" +
    "</html>");
```



Programming languages:

Java

C++

Python

- HTML: Hyper Text Markup Language
  - Mostly used in describe a web page
  - Enclosed with `<html>`…`</html>`

- <h3>: header in level 3
- <p>: paragraph
- <u>, <i>: underline text and *italic* text

Summary

- flow chart
  - Example 1: GCD
    - Building blocks →source code
  - Example 2: Print Triangle Stars
- Looping example
  - While-loop
  - Nested-while-loop
  - Common errors
- Assignment 3 discussion

The end

Thank you!