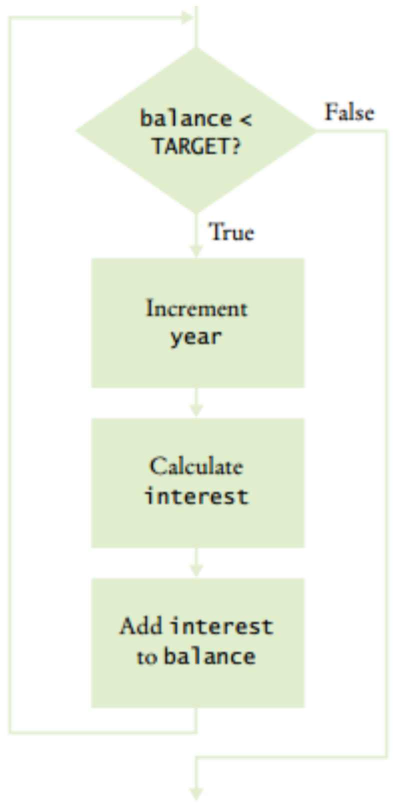
CS 234

Review - Loops

Loops

- While loop
- For Loop
- Do-While
- Nested Loops
- Random numbers

While Loop



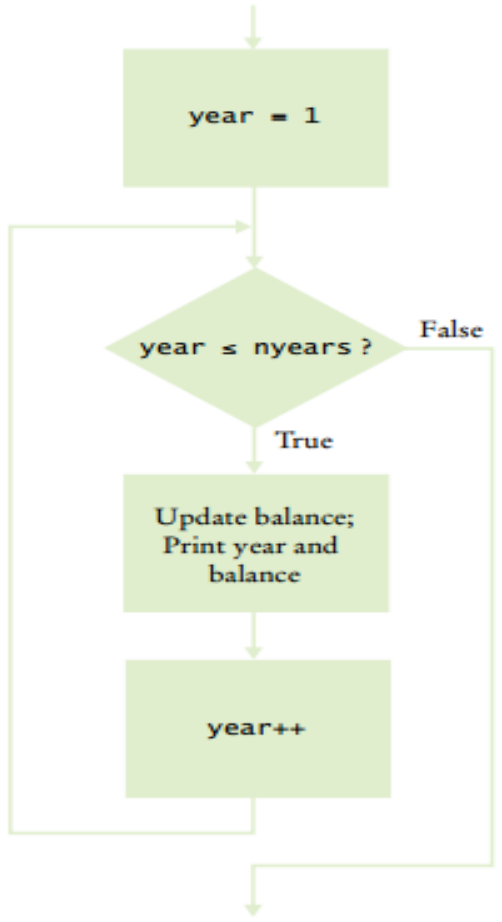
```
public class Main
{
   public static void main(String args[])
   {

       final double RATE = 10;
       final double INITIAL_BALANCE = 10000;
       final double TARGET = 15000;

       double balance = INITIAL_BALANCE;
       int year = 0;

       while (balance < TARGET)
       {
           year++;
           balance = balance * (1+(RATE/100));
       }

       System.out.printf("Target reached in %d years",year);

   }
}
```

While Loop

Hand tracing

Initial balance	target	balance < target?	year	End balance
10,000	15,000	yes	1	11,000
11,000	15,000	yes	2	12,100
12,100	15,000	yes	3	13,310
13,310	15,000	yes	4	14,641
14,641	15,000	yes	5	16,105
16,105	15,000	no	---	---

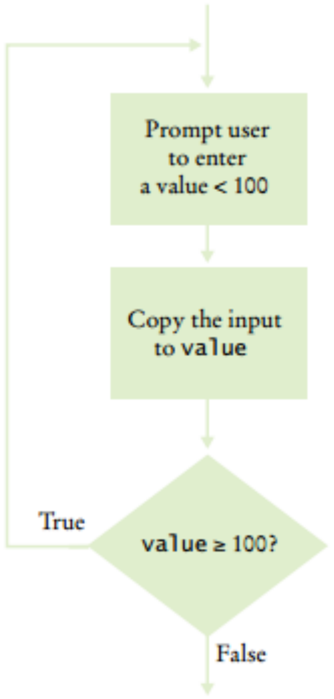
For Loop



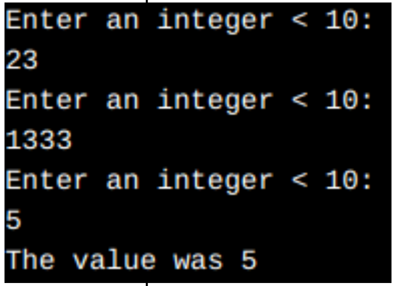
```
public class Main
{
   public static void main(String args[])
   {

       final double RATE = 10;
       final double INITIAL_BALANCE = 10000;
       final int TARGET_YEARS =5;

       double balance = INITIAL_BALANCE;


       for (int year = 1; year <= TARGET_YEARS; year++)
       {
           balance = balance * (1+(RATE/100));
          System.out.printf("%d %.2f\n", year, balance);
       }

       System.out.printf("After %d years I have $%.2f",TARGET_YEARS, balance);
   }
}
```

```
After 5 years I have $16105.10
```

Do While



```
import java.util.Scanner;

public class Main
{
   public static void main(String args[])
   {
        Scanner in = new Scanner(System.in);
        int value;

        do
        {
            System.out.println("Enter an integer < 10:");
            value = in.nextInt();
        }
        while (value >= 10);
        System.out.println("The value was " + value);
   }
}
```

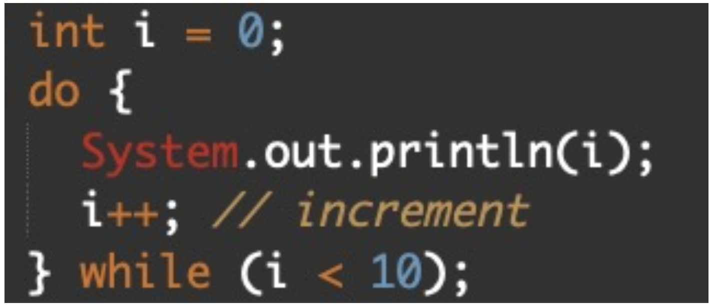


```
Enter an integer < 10:
23
Enter an integer < 10:
1333
Enter an integer < 10:
5
The value was 5
```

It gets executed at least once

Did you notice it?

- While = Do-While = For
  - Initializer
  - Boolean expression
  - Increment operator *



```java
int i = 0;
do {
  System.out.println(i);
  i++; // increment
} while (i < 10);
```

Did you notice it?

- Difference: Initialization, condition, update in one line

```
for (initialization; condition; update)
{
statements
}
```

```
initialization;
while (condition)
{
statements
update
}
```

Which one to use?

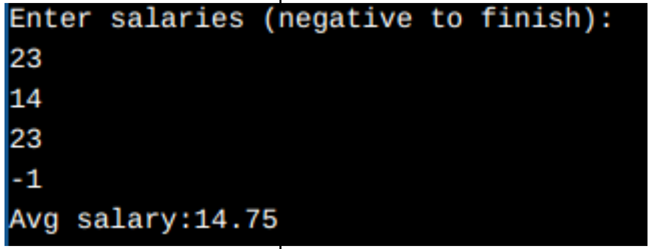
- For Loop
  - Numeric calculation using a variable that is changes by equal amounts each time
  - When you know the number of times to loop
- Do-While
  - If the statements need to be executed at least once
- While
  - If there are circumstances for which the loop body should not be executed at all

Sentinel values

A “special” character or value to signal no more times

E.g., You can use “Q” as quit or -1 in numeric inputs.

```
import java.util.Scanner;

public class Main
{
	public static void main(String[] args) {
	   Scanner in = new Scanner(System.in);
	   double sum = 0;
	   int count = 0;
	   double salary = 0;
	   boolean sentinel = true;
	   System.out.println("Enter salaries (negative to finish):");
	   while (sentinel == true)
	   {
	      salary = in.nextDouble();
	      if (salary < 0)
	         sentinel = false;
	     sum = sum + salary;
	     count++;
	   }
	   if (count > 0)
	   {
	      double average = sum / count;
	      System.out.println("Avg salary:" + average);
	   }
	   else
	   {
	      System.out.println("No data");
	   }
	}
}
```

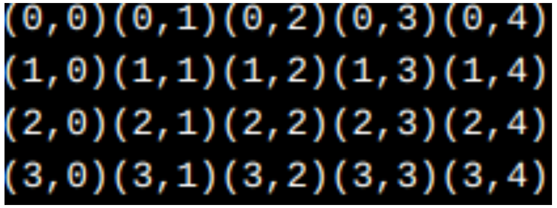


```
Enter salaries (negative to finish):
23
14
23
-1
Avg salary:14.75
```

Nested Loops

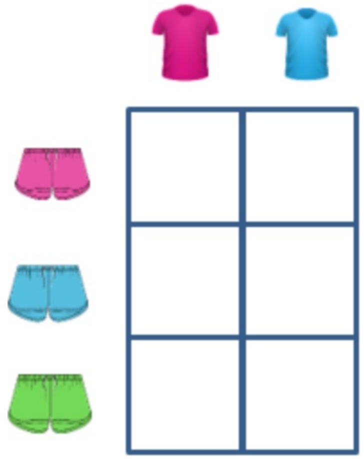
```
public class Main
{
   public static void main(String args[])
   {
        int rows = 3;
        int columns = 4;

        for (int i=0; i<=rows; i++)
        {
            for (int j=0; j<=columns; j++)
            {
                System.out.printf("(%d,%d)",i,j);
            }
            System.out.println();
        }
   }
}
```

What is the output?



```
(0,0)(0,1)(0,2)(0,3)(0,4)
(1,0)(1,1)(1,2)(1,3)(1,4)
(2,0)(2,1)(2,2)(2,3)(2,4)
(3,0)(3,1)(3,2)(3,3)(3,4)
```

Cartesian Product

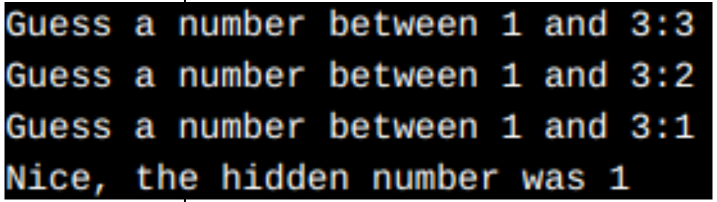


Radom numbers

- Math.random()
- ≥ 0 and < 1
- Trick for getting random integers between a given range
  - (int) (Math.random() * (Upper_number - Lower_number + 1)) + Lower_number
  - E.g., Random number between 1 and 6
  - (int) (Math.random() * (6 – 1 + 1)) + 1

Random numbers

```
import java.util.Scanner;
public class Main
{
   public static void main(String args[])
   {
        Scanner in = new Scanner(System.in);
        final int UPPER = 3;
        final int LOWER = 1;
        int number = (int)(Math.random() * (UPPER - LOWER + 1)) + LOWER;
        int guess;

        do{
            System.out.printf("Guess a number between %d and %d:", LOWER, UPPER);
            guess = in.nextInt();
        }while (guess != number);
        System.out.println("Nice, the hidden number was " + number);
   }
}
```



```
Guess a number between 1 and 3:3
Guess a number between 1 and 3:2
Guess a number between 1 and 3:1
Nice, the hidden number was 1
```