ITEC 120

Lecture 7

Iteration

Review

- Questions
- Design / if / and / or

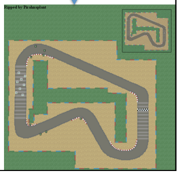
Objectives

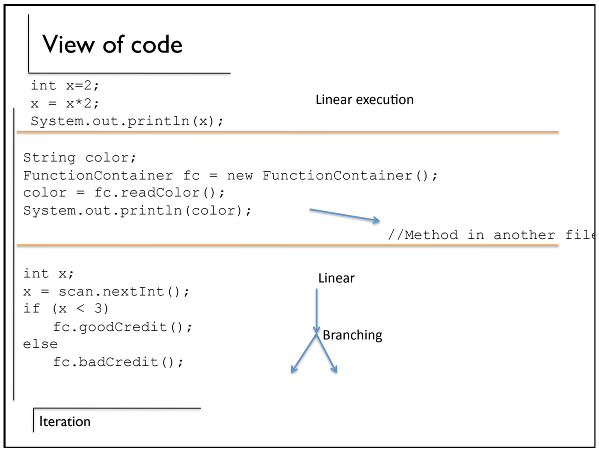
- While loops

Repetition

Without input, this is your program

What if you wanted to do it again?





View of code

```
int x=2;
x = x*2;
System.out.println(x);
```

Linear execution

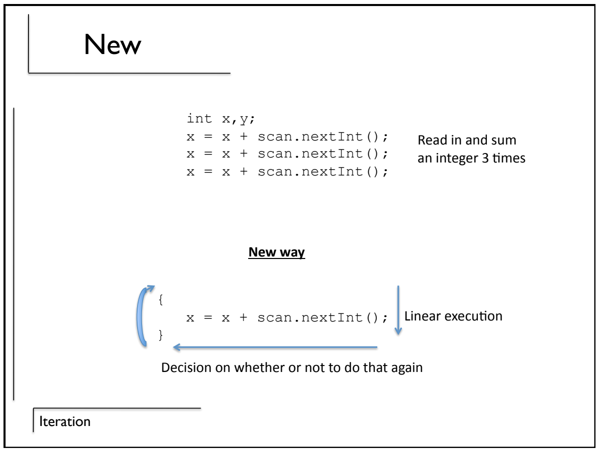
```
String color;
FunctionContainer fc = new FunctionContainer();
color = fc.readColor();
System.out.println(color);
                                    //Method in another fil
```

```
int x;
x = scan.nextInt();
if (x < 3)
    fc.goodCredit();
else
    fc.badCredit();
```

Linear

Branching

Iteration



New

```
int x,y;
x = x + scan.nextInt();
x = x + scan.nextInt();
x = x + scan.nextInt();
```

Read in and sum
an integer 3 times

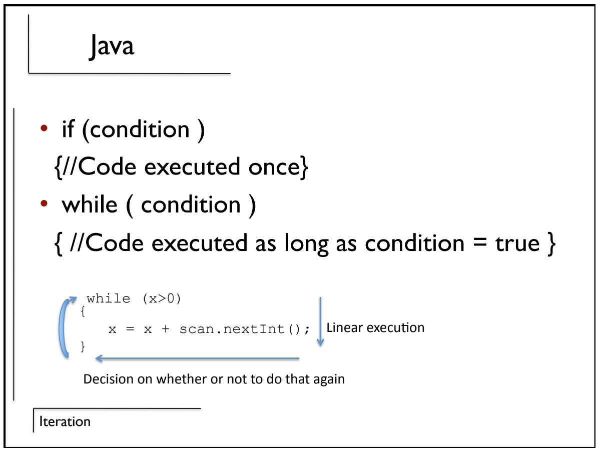
New way

```
{
    x = x + scan.nextInt();
}
```

Linear execution

Decision on whether or not to do that again

Iteration



Java

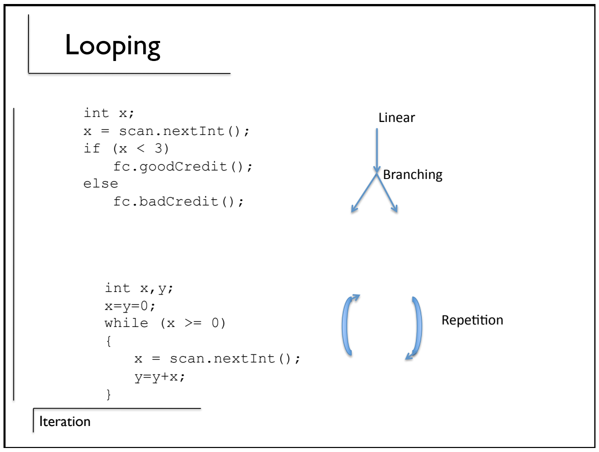
- if (condition)
  {//Code executed once}
- while (condition)
  { //Code executed as long as condition = true }

```
while (x>0)
{
    x = x + scan.nextInt();
}
```

Linear execution

Decision on whether or not to do that again

Iteration



Looping

```
int x;
x = scan.nextInt();
if (x < 3)
    fc.goodCredit();
else
    fc.badCredit();
```

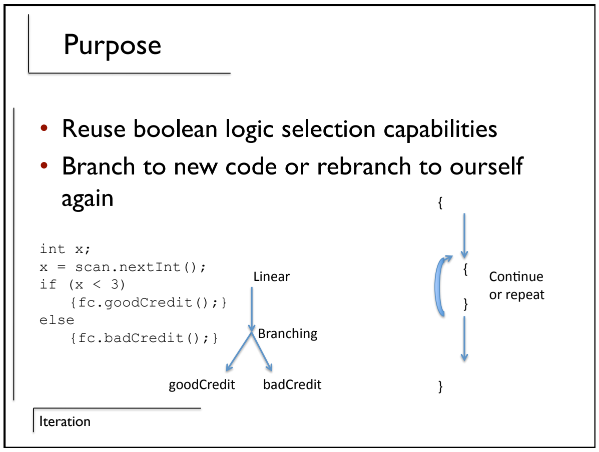
Linear

Branching

```
int x,y;
x=y=0;
while (x >= 0)
{
    x = scan.nextInt();
    y=y+x;
}
```

Repetition

Iteration



Purpose

- Reuse boolean logic selection capabilities
- Branch to new code or rebranch to ourself again

```
int x;
x = scan.nextInt();
if (x < 3)
    {fc.goodCredit();}
else
    {fc.badCredit();}
```

Linear

Branching

goodCredit

badCredit

{

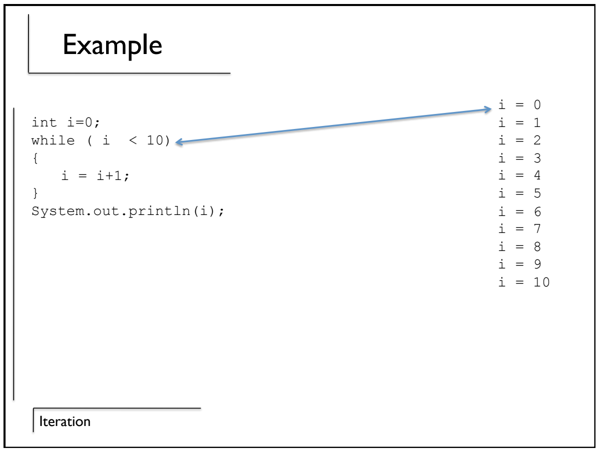
{

}

}

Continue or repeat

Iteration

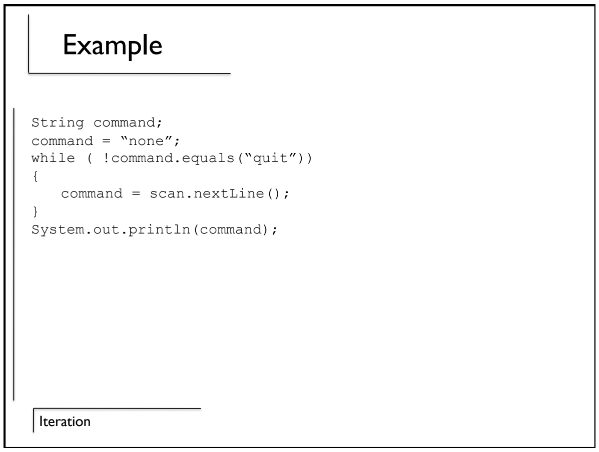


Example

```
int i=0;
while ( i  < 10)
{
    i = i+1;
}
System.out.println(i);
```

```
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
```

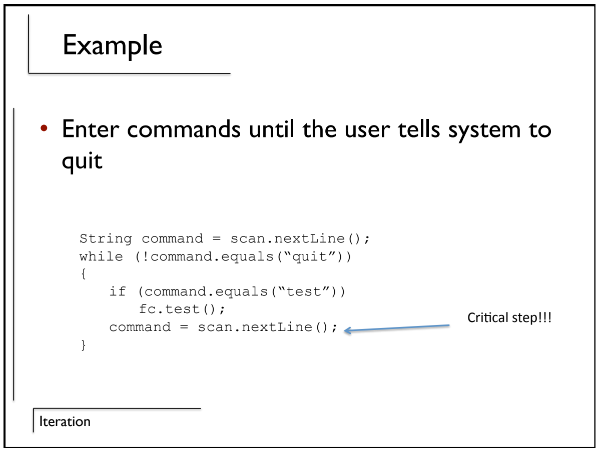
Iteration



Example

```
String command;
command = “none”;
while ( !command.equals(“quit”))
{
    command = scan.nextLine();
}
System.out.println(command);
```

Iteration



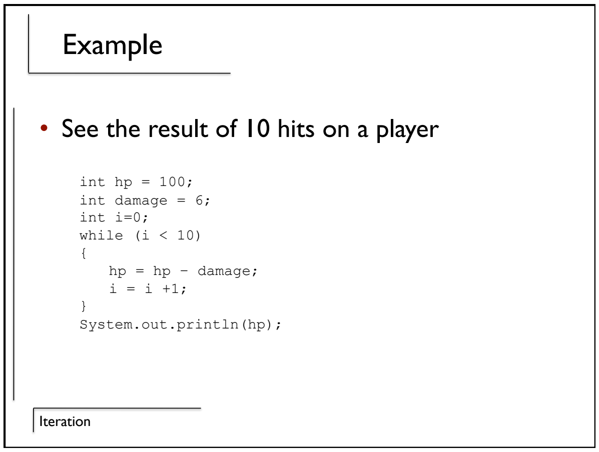
Example

- Enter commands until the user tells system to quit

```
String command = scan.nextLine();
while (!command.equals(“quit”))
{
    if (command.equals(“test”))
        fc.test();
    command = scan.nextLine();
}
```

Critical step!!!

Iteration

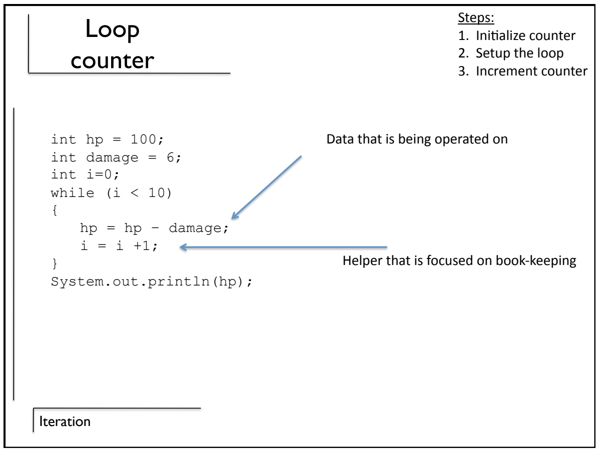


Example

- See the result of 10 hits on a player

```
int hp = 100;
int damage = 6;
int i=0;
while (i < 10)
{
   hp = hp - damage;
   i = i +1;
}
System.out.println(hp);
```

Iteration



Loop counter

Steps:

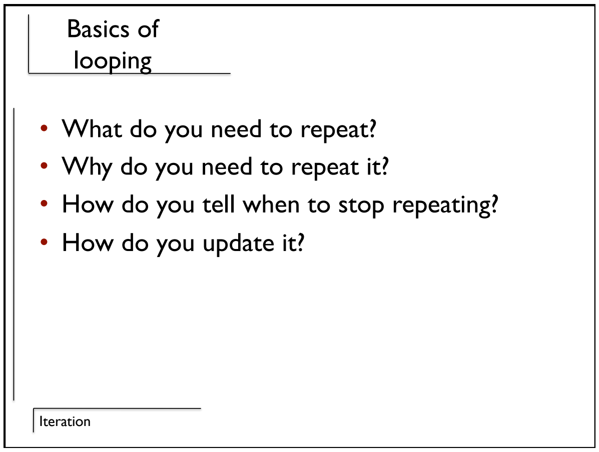
1. Initialize counter
2. Setup the loop
3. Increment counter

```
int hp = 100;
int damage = 6;
int i=0;
while (i < 10)
{
   hp = hp - damage;
   i = i +1;
}
System.out.println(hp);
```

Data that is being operated on

Helper that is focused on book-keeping

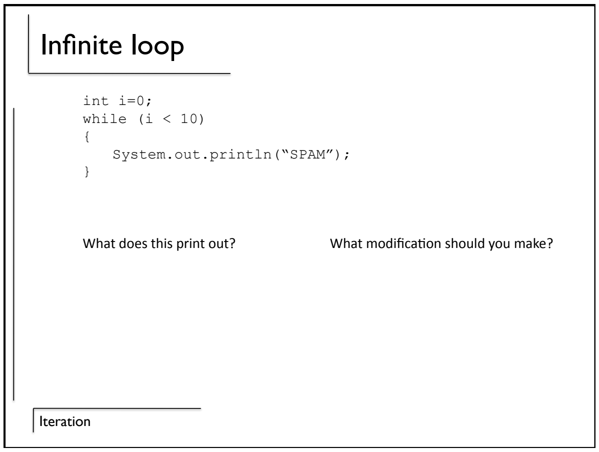
Iteration



Basics of looping

- What do you need to repeat?
- Why do you need to repeat it?
- How do you tell when to stop repeating?
- How do you update it?

Iteration



Infinite loop

```
int i=0;
while (i < 10)
{
   System.out.println(“SPAM”);
}
```

What does this print out?

What modification should you make?

Iteration

Problems

- Sum numbers until user types in -1
- Interactive menu with commands

Iteration

Breaking out

- What if you get cold feet halfway through?
- How do you tell a loop to stop executing?
- Two statements to modify behavior of loops
  - continue
  - break

Iteration

break

What is the counter variable?
What is the sentinel?
How is it updated?

- Exits loop at that point

```
while(true)
{
    String command = scan.nextLine();
    if (command.equals(“process”))
    {
        fc.process();
    }
    else if (command.equals(“quit”))
    {
        break;
    }
}
```

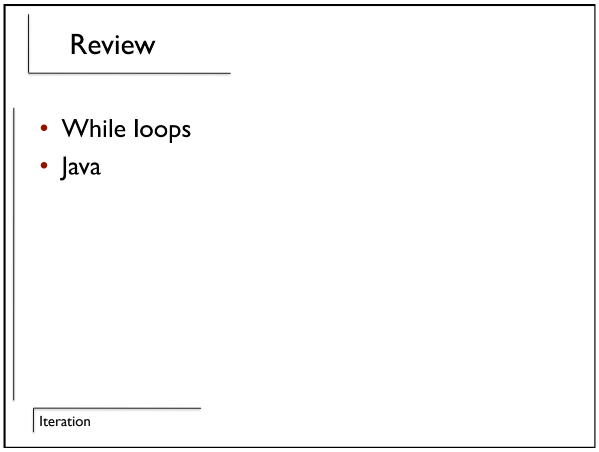
Iteration

Continue

- Allows you to stop a loop, begin at the top

```
while(true)
{
    String command = scan.nextLine();
    if (command.equals(“wait”))
    {
        continue;
    }
    else if (command.equals(“move”))
    {
        fc.move();
    }
}
```

Iteration



Review

- While loops
- Java

Iteration