

# Lesson 36: for() Loops (W11D1)

Balboa High School

Michael Ferraro

October 26, 2015

- Create a new project: Lesson36
- Write class FirstForLoop:
  - Include a main() method:

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

- Implement printEvensToTen():

```
public static void printEvensToTen() {  
    //using a while() loop, print the even  
    //numbers from 2 to 10, inclusive...  
}
```

Students will learn how to write and use `for()` loops and work on §5 programming exercises from PS #5.

## printEvensToTen() via Modulo (%)

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i <= 10 ) {  
  
        if ( i % 2 == 0 ) { //even  
            System.out.println(i);  
        }  
        i++;  
  
    }  
  
}
```

## printEvensToTen(), Another Way

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i <= 10 ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}
```

## printEvensToTen(), Another Way

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i <= 10 ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}  
  
//what other stopping conditions would work?
```

## printEvensToTen(), Another Way

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i < 11 ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}  
  
//and...
```

## printEvensToTen(), Another Way

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i < 12 ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}
```



# Important Parts of a while() Loop

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i <= 10 ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}
```

## Initialization

## Important Parts of a while() Loop

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i <= 10 ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}
```

**Initialization**      **Stopping Condition**

## Important Parts of a while() Loop

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    while ( i <= 10 ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}
```

**Initialization**    **Stopping Condition**    **Loop Variant**

## Another Look at for()

- We've used for() loops before:

```
ArrayList<String> bones =  
    new ArrayList<String>();  
  
bones.add("femur");  
bones.add("fibula");  
bones.add("tibia");  
  
for( String bone : bones ) {  
    System.out.println( bone );  
}
```

This is the *for-each* or *enhanced* form of the for() loop

## Another Look at for()

- The other use for a for() loop is to function much like a while() loop.
- Update printEvensToTen():

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    for ( ; i <=10 ; ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
  
}
```

# Syntax of for()

- Recall the important aspects of a while() loop:
  - **Initialization**
  - **Stopping Condition**
  - **Loop Variant**

# Syntax of for()

- Recall the important aspects of a while() loop:
  - **Initialization**
  - **Stopping Condition**
  - **Loop Variant**
- Syntax of a for() loop:

```
for( init'n ; stopping cond'n ; loop variant ) {...}
```

# Syntax of for()

- Recall the important aspects of a while() loop:
  - **Initialization**
  - **Stopping Condition**
  - **Loop Variant**
- Syntax of a for() loop:

```
for( init'n ; stopping cond'n ; loop variant ) {...}
```

- Any part may be omitted, as you saw in our updated printEvensToTen()



## Making Better Use of for()

- Update printEvensToTen() again...

```
public static void printEvensToTen() {  
  
    int i = 2;  
  
    for ( ; i <=10 ; ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
}
```

## Making Better Use of for()

- Update printEvensToTen() again...

```
public static void printEvensToTen() {  
  
    int i;  
  
    for ( i = 2 ; i <=10 ; ) {  
  
        System.out.println(i);  
        i += 2;  
  
    }  
}
```

## Making Better Use of for()

- Update printEvensToTen() again...

```
public static void printEvensToTen() {  
  
    int i;  
  
    for ( i = 2 ; i <=10 ; i += 2 ) {  
  
        System.out.println(i);  
  
    }  
}
```

## Making Better Use of for()

- Even the declaration of `i` can be moved:

```
public static void printEvensToTen() {  
  
    for ( int i = 2 ; i <=10 ; i += 2 ) {  
  
        System.out.println(i);  
    }  
}
```

This is much more compact than a `while()` loop!

# An Important Difference

- How high does i get?

```
public static void printEvensToTen() {  
  
    for ( int i = 2 ; i <=10 ; i += 2 ) {  
  
        System.out.println(i);  
    }  
  
    System.out.println("i = " + i);  
}
```

Try this out and see if you can identify the problem.

# An Important Difference

- How high does i get?

```
public static void printEvensToTen() {  
  
    for ( int i = 2 ; i <=10 ; i += 2 ) {  
  
        System.out.println(i);  
    }  
  
    System.out.println("i = " + i);  
}
```

Conclusion: Vars declared inside `for()` loop have scope within the loop only.

# Exercise #1

Inside `main()`, write an equivalent set of statements using `for()` that mimics the following:

```
int a = 0;

while ( a <= 6 ) {
    System.out.println("a is " + a);
    a++;
}
```

## Exercise #2

Inside `main()`, write an equivalent set of statements using `for()` that mimics the following:

```
int n = 6;
int a = 0;

while ( a < n ) {
    System.out.println("a is " + a);
    a += 2;
}
```



## Exercise #3

Inside `main()`, write an equivalent set of statements using `for()` that mimics the following:

```
int i = 10;

while ( i > 0 ) {
    System.out.println("i is " + i);
    i--;
}
```

## Exercise #4

Inside `main()`, write an equivalent set of statements using `for()` that mimics the following:

```
int i = 10;

while ( i > 0 ) {
    System.out.println("i is " + i);
    i--;
}

System.out.println("i reaches " + i);
```

Continue working on §5 of PS #5 — make sure you're done by next class!