

# Switch Statement

```
String data = "fall";
switch(data) {
    case "fall":
        // statements
        break;
    case "winter":
        // statements
        break;
    case "spring":
        // statements
        break;
    default:
        // these statements execute if all the cases failed
        break;
}
```

# Switch Statement

- Supported data types:
  - byte
  - short
  - char
  - int
  - String
  - enum types
  - wrapper classes

# String equality

How can we tell that 2 strings are equal?

```
String str1 = "hello";  
String str2 = "hello";  
if (str1 == str2) // WRONG!!!  
    ...
```

# String equality

```
String str1 = "hello";  
String str2 = "hello";  
if (str1.equals(str2)) // Correct 😊  
    ...
```

```
// Prompt user for password

final String PASSWORD = "123456";
...
System.out.println("Please enter your password.");
String password = keyboard.next();

if(password.equals(PASSWORD))
    System.out.println("Permission granted.");
else
    System.out.println("Permission denied.");
```

# String Comparisons

*stringObject.compareTo(stringArgument)*

Returns: <0 , 0 , or >0

	0	1	2	3	4	5	6	7
studentName	K	a	y	,	_	J	o	
teacherName	K	a	y	,	_	A	m	y
<b>studentName &gt; teacherName</b>								<i>studentName &gt; teacherName evaluates to true</i>
<i>Each comparison uses ASCII values</i>	75	97	121	44	32	74		
	75	97	121	44	32	65		
	=	=	=	=	=	>		

# Other String Methods

*length()*

*isEmpty()*

*indexOf(charVal)* - **charVal** is a char

*indexOf(charVal, index)* – starting at **index**

*substring(startIndex, endIndex)* – starting and ending indices of substring

*equalsIgnoreCase(str)* – compares equality of two strings ignoring case

# String Modifier Methods

`concat(stringArg)` – ex: `str1.concat(str2)`  
is equivalent to `str1 + str2`

`replace(str1, str2)` – finds **str1**, replaces with **str2**

`replace(char1, char2)` – finds **char1**, replaces with **char2**

`str1 += str2` – equivalent to `str1 = str1 + str2`



# Character Operations

- More static methods (recall, the Math methods are static methods)

`isLetter(charArg)` – ex:

`Character.isLetter('c')` returns true

`isDigit(charArg)`

`isWhitespace(charArg)`

`toUpperCase(charArg)`

`toLowerCase(charArg)`

# System.out.printf()

```
int quant = 25;  
double price = 4.99;  
System.out.printf("%d hats at %f", quant, price);
```

Output:

```
25 hats at 4.990000
```

Use /n for a new line

```
//What if we want to format our double to 2 sig figs?  
int quantity = 25;  
double price = 4.99;  
System.out.printf("%d hats at %.2f", quant, price);
```

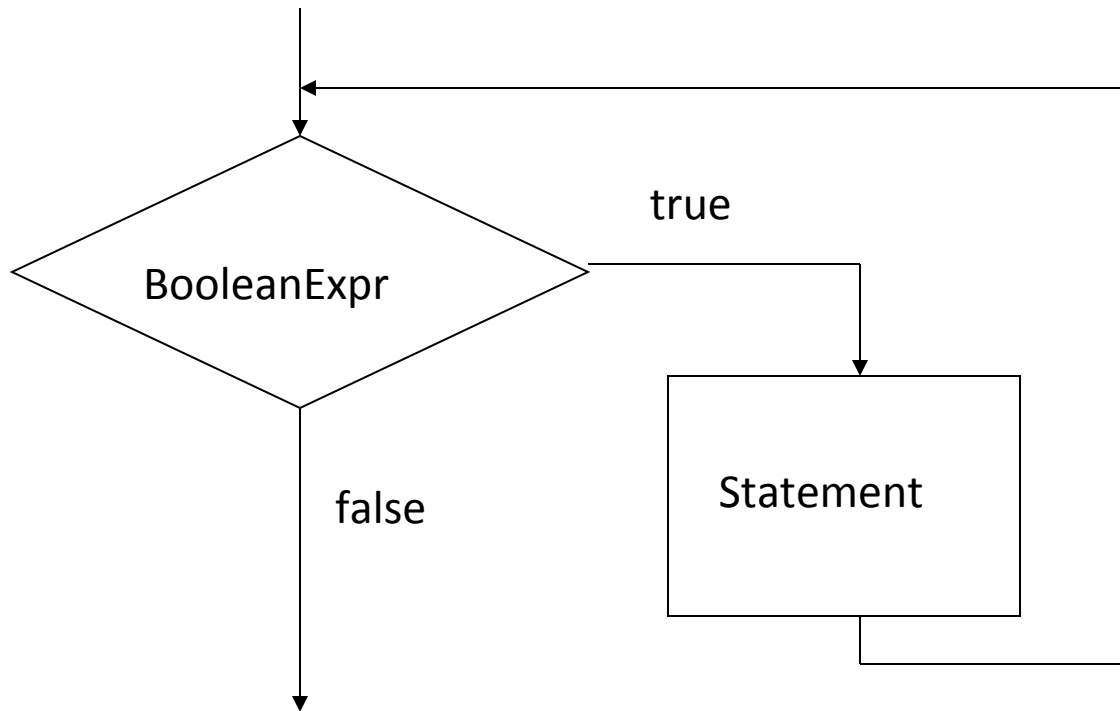
Output:

```
25 hats at 4.99
```

# The while Loop

`while ( BooleanExpression )`  
`Statement`

Execution enters the loop



Continue with the rest of the program

# General form of a while loop

```
while ( BooleanExpression ) {  
    Body  
}
```

# More Specifically

```
InitializationStatement  
while ( BooleanExpression ) {  
    Statement1  
    Statement2  
    ...  
    PrepareForNextIteration  
}
```

# Average.java

```
// Enter zero when finished
System.out.println("Enter numbers for average.");
userNum = keyboard.nextInt();
while(userNum != 0) {
    sum += userNum;
    quant++;
    userNum = keyboard.nextInt();
}
System.out.println("You entered " + quant +
    " numbers, with an average of " +
    ((double)sum/quant));
```

# While loop practice

// How many times does the while loop iterate below?

```
int counter = 0;
while(counter < 10){
    counter++;
}
```

# While loop practice

// How many times does the while loop iterate below?

```
int counter = 1;
while(counter != 10){
    counter += 2;
}
```



```
// Prompt user for password, they only get 3 tries.

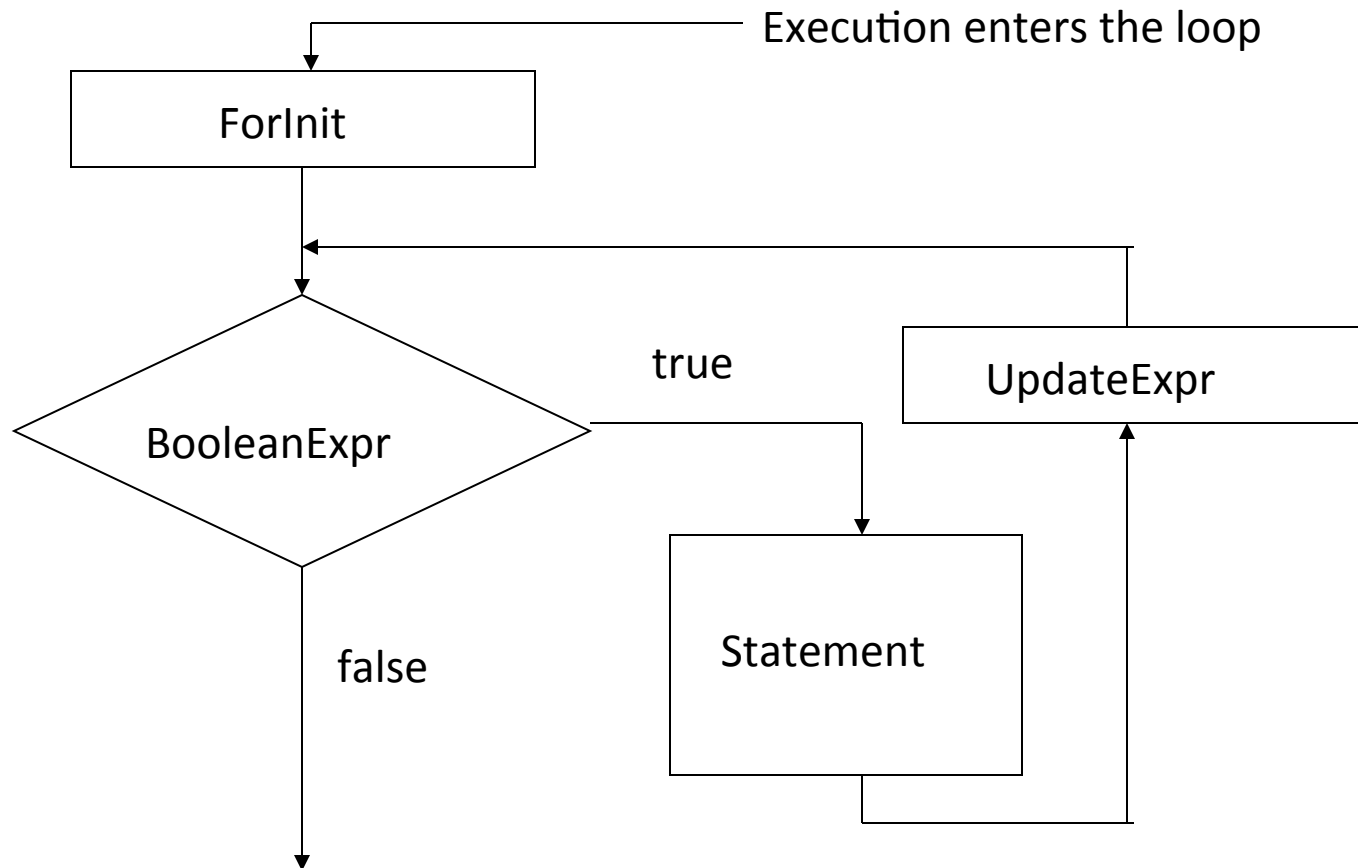
System.out.println("Please enter your password.");
String password = keyboard.next();
int numTries = 1;

while(!password.equals(PASSWORD) && (numTries <= 2))
{
    System.out.println("Access Denied. Try again.");
    password = keyboard.next();
    numTries++;
}

if(numTries > 2)
    System.out.println("Access granted.");
else
    System.out.println("Hold on account.");
```

# The for Loop

`for ( ForInit; BooleanExpr; UpdateExpr )`  
*Statement*



Continue with the rest of the program

```
// Simple for loop
```

```
for(int i = 0; i < 10; i++){  
    System.out.println("Value of i: " + i);  
}
```

Output:

```
Value of i: 0  
Value of i: 1  
Value of i: 2  
Value of i: 3  
Value of i: 4  
Value of i: 5  
Value of i: 6  
Value of i: 7  
Value of i: 8  
Value of i: 9
```

// Convert the following while loop to a for loop

```
int count = 1;
while(count <= 10){
    System.out.println("count value: " + count);
    count++;
}
```

```
//Savings account calculator. Assume 5% interest/year

System.out.println("Starting balance:");
double balance = keyboard.nextDouble();
System.out.println("Years drawing interest?");
int years = keyboard.nextInt();

for(int i = 0; i < years; i++){
    balance = balance*1.05; // 5% interest
}

System.out.printf("Balance after %d is $%.2f", years,
                 balance);
```

# do-while

Like a while loop, but the body will always execute at least once.

```
do {  
    statement(s)  
} while (expression);
```