

< Conditionals

- if
- Conditional operator
- switch

< True/False

- Condition using conditional operator
- Condition using logical operator
- Conditional enclosed within parens
- Tested for true or false
-

< if statement

- if keyword
- Boolean expression enclosed within parens
- Code block (statement or block of statements to executed if boolean expression is true)

```
if (boolean expression)
{
block of code to be executed if true
}
```

< String Testing

- Using == with Strings determines whether the reference variables are pointing to the same location (same String)

- To determine whether the 2 reference variables are pointing to different locations that contain the same values, use equals method

< example1

```
1 public class StringTest
2 {
3     public static void main (String args[])
4     {
5         String str1, str2, str3;
6         str1 = "George Washington";
7         str2 = "John Adams";
8         if (str1.equals(str2));
9     {
10        System.out.println("Same name.");
11    }
12 }
13 }
```

< example2

```
1 public class StringTest
2 {
3     public static void main (String args[])
4     {
5         String str1, str2, str3;
6         str1 = "George Washington";
7         str2 = "John Adams";
8         if (str1==str2);
9     {
10        System.out.println("Same String.");
11    }
12 }
13 }
```

< if, else

```
The if, else statement syntax:  
If (boolean expression)  
{  
statement(s) to be executed if true;  
}  
else  
{  
statement(s) to be executed if false;  
}
```

< Style

- Enclosed the blocks to be executed in braces, even if the block contains only one statement
- Indent the code and braces
- Enter the first brace, then the closing brace
- Go back and enter the block of code to be executed

< Nested if Statements

- multiple factors before making a selection
- Example: letter grade for numerical test scores.

< switch statement

- Instead of nested if, else construct, might use
- switch statement for multiple conditions
- However, use when the boolean
- expressions:

- Test the value of an int, short, byte, or
- char type
- Are equality tests
- Are tests against a single variable

< Syntax for switch Constructs

- variable- can be only char, byte, short, or int
- case labels – keyword to identify when equality occurs
- literal_value – match against the expression tested
- code_block – The actions to perform if equal
- break – goes to end of switch statement
- default – executed code if no equality
-

< switch statement syntax

```
Switch (expr1)
{
case constant 2:
Statements;
Break;
case constant 3:
Statements;
Break;
default:
Statements;
Break
}
```

< Conditional ternary Operator

- Quick test of an expression
- Syntax:

- (boolean exp) ? Value if true : value if false;
- Shipping = (total>50) ? 0 : 10;