Returning Within a Conditional

Returning Within a Conditional

- When a **return statement** is reached, the specified **value is returned**, and we **exit the method**
  - Any remaining portion of the method is NOT executed

- **Must return on ALL paths out of a method**

Recap – Conditionals

```
if (<condition>) {
    <controlled stmt(s)>;
}
<statement(s)>;
```

Zero or One
of these sets of controlled stmts will be executed

```
if (<cond1>) {
    <cond1 controlled stmt(s)>;
} else if (<cond2>) {
    <cond2 controlled stmt(s)>;
}
<statement(s)>;
```

```
if (<condition>) {
    <if controlled stmt(s)>;
} else {
    <else controlled stmt(s)>;
}
<statement(s)>;
```

Exactly One
of these sets of controlled stmts will be executed

```
if (<cond1>) {
    <cond1 controlled stmt(s)>;
} else if (<cond2>) {
    <cond2 controlled stmt(s)>;
} else {
    <else controlled stmt(s)>;
}
<statement(s)>;
```

Q: Will the following code compile and run?

```
public class ReturnConditional {
    public static void main(String[] args) {
        System.out.println(isPositive(-2));
    }

    public static String isPositive(int a) {
        if (a > 0 ) {
            return "Positive";
        } else if ( a < 0 ) {
            return "Negative";
        } else if ( a == 0 ) {
            return "Zero";
        }
    }
}
```

Zero or One of these sets of controlled statements will be executed

Given the conditional structure, the compiler believes it is possible that none of the return statements will be reached.

```
$ javac ReturnConditional.java
ReturnConditional.java:15: error: missing return statement
    }
    ^
1 error
```

Corrected implementation:

```
public class ReturnConditional {
    public static void main(String[] args) {
        System.out.println(isPositive(-2));
    }

    public static String isPositive(int a) {
        if (a > 0 ) {
            return "Positive";
        } else if ( a < 0 ) {
            return "Negative";
        } else {
            return "Zero";
        }
    }
}
```

```
$ javac ReturnConditional.java
$ java ReturnConditional
Negative
```

More examples (equivalent implementations)

```java
/**
 * Returns the max of x and y
 * @param x integer to compare
 * @param y integer to compare
 * @return the max of x and y
 */
public static int maxA(int x, int y) {
    int max = y;
    if (x > y) {
        max = x;
    }
    return max;
}
```

```java
public static int maxB(int x, int y) {
    if (x > y) {
        return x;
    } else {
        return y;
    }
}
```

```java
public static int maxC(int x, int y) {
    if (x > y) {
        return x;
    }
    return y;
}
```