

Conditionals

Why a Conditional?

As we write more complicated programs, we don't always want them to do the same thing in all cases. We might want them to do one thing in one situation, and another thing in a different one. For example, a computer game does one thing while the game is in progress, but behaves differently when the game is over.

If-Statements

An if-statement allows us to do something only if some condition is true. There are also variants on if-statements that allow us to do one thing in one case, another thing in another case, etc.

Simple If-Statement

Here is the syntax for a simple if-statement:

```
if (condition) {  
    //statements  
}
```

Here, `condition` is either a boolean variable or an expression that evaluates to true or false. The `statements` inside the if-statement are things that we ONLY want to happen if our condition is true. Finally, the brackets `{ }` around the body are optional if there is only one statement inside.

For example, suppose we want to get the user's name and age as input. In any case, we want to then print out a greeting. We also want to print "You are an adult" if the user is 18 or older. Here is the full program:

```
import java.util.*;  
public class HelloAdult {  
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
  
        System.out.print("Enter your name: ");  
        String name = s.nextLine();  
        System.out.print("Enter your age: ");  
        int age = Integer.parseInt(s.nextLine());  
  
        System.out.println("Hello, " + name);  
        if (age >= 18) {  
            System.out.println("You are an adult");  
        }  
    }  
}
```

When the program runs, it will ask the user for their name and age and read those values into the name and age variables. It will then print Hello and the person's name. Finally, if the person is 18 or older, the program will print "You are an adult". If the user is under 18, the program will not print anything else.

If/Else-Statement

Here is the syntax for an if/else-statement. This allows us to do one thing if something is true, and a different thing if it is not true:

```
if (condition) {
    //statements
}
else {
    //statements
}
```

Here, we will check the condition and execute the first set of statement if the condition is true, and the second set of statements if the condition is not true. In any case, exactly one set of statements is executed.

For example, suppose we want to print out whether a number is even or odd. We can tell if a number is even by checking to see if it is divisible by two.

```
Scanner s = new Scanner(System.in);
System.out.print("Enter a number: ");
int num = Integer.parseInt(s.nextLine());

//If the remainder of num/2 is 0, then num is even
if (num % 2 == 0) {
    System.out.println(num + " is even");
}
else {
    System.out.println(num + " is odd");
}
```

(Note: this is not a complete program. We would need to put this code inside a class declaration and inside the main method.) When this code is executed, it will first ask the user for a number, and store the result in the num variable. It will then compute num%2 (the remainder of num/2). If it is 0, then it will print that the number is even. If it is anything else, it will print that the number is odd. Each time, only one of the statements will be printed.

If/Else If-Statement

Here is the syntax for an if/else if-statement. This allows us to do different things for several different cases:

```
if (condition1) {
    //statements
}
else if (condition2) {
    //statements
}
...
else {
    //statements
}
```

In this structure, we first evaluate `condition1`. If it is true, then we execute the first set of statements and leave the if/else if statement. If `condition1` is false, then we step down and evaluate `condition2`. If it is true, then we execute the second set of statements and move on in the program. If it is false, we move down to evaluate the next condition. This process continues until we run out of conditions to check. If none of the conditions were true, then the statements inside the “else” are executed.

We can have as many “else if” conditions as we want in this structure (which is denoted by the `...`). Furthermore, the “else” portion is optional – we don’t have to have a special section that executes if none of the conditions were true.

For example, suppose we want to print out an age category for the user:

```
Scanner s = new Scanner(System.in);
System.out.print("Enter your age: ");
int age = Integer.parseInt(s.nextLine());

if (age <= 12) {
    System.out.println("You are a child");
}
else if (age < 19) {
    System.out.println("You are a teenager");
}
else {
    System.out.println("You are an adult");
}
```

In this example, exactly one of the statements will be printed, depending on the user’s age.

More Examples

As another example, suppose you want to ask the user for 5 tests grades (out of 100) so that you can calculate their overall letter grade (90% and up is an A, 80-90% is a B, etc.) Here is a fragment of the program:

```
Scanner s = new Scanner(System.in);
System.out.print("Enter a test score: ");
int grade1 = Integer.parseInt(s.nextLine());

System.out.print "Enter a test score: ";
int grade2 = Integer.parseInt(s.nextLine());

System.out.print("Enter a test score: ");
int grade3 = Integer.parseInt(s.nextLine());

System.out.print("Enter a test score: ");
int grade4 = Integer.parseInt(s.nextLine());

System.out.print("Enter a test score: ");
int grade5 = Integer.parseInt(s.nextLine());

double avg = (grade1+grade2+grade3+grade4+grade5)/5.0;

if (avg >= 90.0) {
    System.out.println("Letter grade: A");
}
else if (avg >= 80.0) {
    System.out.println("Letter grade: B");
}
else if (avg >= 70.0) {
    System.out.println("Letter grade: C");
}
else if (avg >= 60.0) {
    System.out.println("Letter grade: D");
}
else {
    System.out.println("Letter grade: F");
}
```

If-statements are also very useful in handling error conditions in programs. You can check to see if the user's input is what you expected before performing any calculations. For example, in the grade calculator we just wrote, we expect the test scores to be between 0 and 100. As an error condition, we could add this check after getting all the user input:

```
Scanner s = new Scanner(System.in);
if (grade1<0 || grade1>100 || grade2<0 || grade2>100 || grade3<0 ||
    grade3>100 || grade4<0 || grade4>100 || grade5<0 || grade5>100) {
```

```

        System.out.print("Invalid input");
    }
    else {
        //put the avg calculation and the code to print the letter grade here
    }
}

```

Now, a letter grade will only be printed if all test scores had appropriate values.

Switch Statements

There is a second type of conditional statement in Java called a *switch statement*. These statements can be used as a shortcut over if-statements when checking the value of variables or expressions.

Syntax

Here is the syntax of a switch statement:

```

switch (expression) {
    case val1:
        //statements
        break;
    case val2:
        //statements
        break;
    ...
    default:
        //statements
}

```

Here, `expression` is either an expression or variable that evaluates to a `char` or an `int`. Inside the statement, `val1`, `val2`, etc. are possible values for `expression`. If the expression evaluates to `val1`, then we execute the statements inside the `val1` case. If the expression evaluates to `val2`, then we execute the statements inside the `val2` case. We can have as many cases as we want (as denoted by the `...`). The `default` case is executed if the expression does not evaluate to any of the case values. This `default` case is optional.

The “`break`” statements at the end of each case mean that we will leave the switch statement at the end of a case. They are also optional, but if we leave them out then we will continue on to the statements in the next case (WITHOUT checking the case value). We will see an example of this in the next sections.

Examples

Here is a simple example that uses a switch statement to print out comments about a given letter grade:

```

Scanner s = new Scanner(System.in);

```

```

System.out.print("Enter your letter grade");

//this technique converts the string to a char
char grade = (s.nextLine()).charAt(0);

switch (grade) {
    case 'A':
        System.out.println("Excellent");
        break;
    case 'B':
        System.out.println("Good");
        break;
    case 'C':
        System.out.println("Average");
        break;
    case 'D':
        System.out.println("Poor");
        break;
    case 'F':
        System.out.println("Failure");
        break;
    default:
        System.out.println("Invalid grade");
        break;
}

```

Depending on what grade the user entered, exactly one of the case statements will be printed.

To see what happens when we leave some break statements out, suppose we want to repeat the example, but just print out whether the user passed (A, B, or C) or failed (D or F). Here's what we would do:

```

Scanner s = new Scanner(System.in);
System.out.print("Enter your letter grade");
char grade = (s.nextLine()).charAt(0);

switch (grade) {
    case 'A':
    case 'B':
    case 'C':
        System.out.println("Pass");
        break;
    case 'D':
    case 'F':
        System.out.println("Fail");
        break;
    default:

```

```

        System.out.println("Invalid grade");
    }

```

If the user enters 'A', for example, then we will first go to case 'A' in the switch statement. Since there is no break statement, we will go directly to case 'B' (even though the grade entered doesn't match the case). There is not a break statement in 'B' either, so we will go to case 'C'. There we will print "Pass" (which we should for an 'A' grade) and break out of the switch statement.

Full Examples

This section includes three examples of full programs using conditional statements.

Example 1

There are many different formulas for estimating a person's maximum heart rate using their age and gender. Here is one such estimation:

For men: $220 - \text{age}$
 For women: $206 - 88\% \text{ of age}$

(Every such formula is only an estimation, and may or may not be accurate for a particular person.) Below is a complete program that asks for a user's age and gender, and then prints their maximum heart rate according to the estimation above.

```

import java.util.*;

public class Example1 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        System.out.print("Enter your age: ");
        int age = Integer.parseInt(s.nextLine());
        System.out.print("Enter (m)ale or (f)emale: ");
        char gender = (s.nextLine()).charAt(0);

        if (gender == 'm' || gender == 'M') {
            int max = 220-age;
            System.out.println("Max HR: " + max);
        }
        else if (gender == 'f' || gender == 'F') {
            double maxF = 206 - 0.88*age;
            System.out.println("Max HR: " + maxF);
        }
        else {
            System.out.println("Invalid gender");
        }
    }
}

```

```
}
```

Example 2

In this example, we will get two numbers as input from the user, and we will print out whichever number is smaller. In the case of a tie, it doesn't matter which number we print.

```
import java.util.*;

public class Example2 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int num1 = Integer.parseInt(s.nextLine());
        System.out.print("Enter the second number: ");
        int num2 = Integer.parseInt(s.nextLine());

        if (num1 < num2) {
            System.out.println(num1);
        }

        else {
            System.out.println(num2);
        }
    }
}
```

Example 3

In this example, we will ask the user for a temperature and the system being used (Fahrenheit or Celsius). We will then print the equivalent temperature in the other system (i.e., if we start with Fahrenheit then we will convert to Celsius, and vice versa). Here are the formulas that are used to convert between the two (C is Celsius and F is Fahrenheit):

$$C = (5/9)*(F-32)$$
$$F = (9/5)*C+32$$

```
import java.util.*;

public class Example3 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);

        System.out.print("Enter the temperature: ");
        double temp = Double.parseDouble(s.nextLine());
        System.out.print("Enter (f)ahrenheit or (c)elsius: ");
```



```
char system = (s.nextLine()).charAt(0);

double newTemp;

if (system == 'f' || system == 'F') {
    newTemp = (5.0/9.0)*(temp-32);
}
//we are assuming that the user typed either f or c
else {
    newTemp = (9.0/5.0)*temp + 32;
}

System.out.println("Converted: " + newTemp);
}
}
```