2.2 - Making Decisions

So far we have only made programs that execute the statements in order, starting with the statements at the top of the screen and moving down. However, you can write programs that can change the order that statements are executed based on true and false conditions.

The Boolean Datatype

Boolean variables can only be true or false, both of which are keywords in Java. Boolean variables can be used to store the outcome of boolean expressions.

Boolean Expressions

In Java, you can create boolean expressions with the comparison operators (somtimes called relational operators).

Operator	What it Means
>	greater than
<	less than
>=	greater or equal to
<=	lesser or equal to
==	equal to
!=	not equal to

The if Statements

Although you can store the value of a boolean expression in a variable, the most common place to use boolean expressions are in conditional statements such as the if statement:

```
if (boolean expression)
{
    statements;
}
```

Here the *statements* will be executed only if the *boolean expression* is true. If it is false, java will skip the *statements* and go to the first statement after the if.

Example Program:

```
import java.util.Scanner;

class MagicNumber
{
    public static void main(String[] args)
    {
        final int MAGIC_NUMBER = 42;
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the magic number: ");
        int guess = sc.nextInt();
        if (guess == MAGIC_NUMBER)
        {
            System.out.println("You guessed it!");
        }
        System.out.println("You failed.");
    }
}
```

In this program, Java will only print "You guessed it!" if you correctly guess the magic number. Otherwise it will always print "You failed." Make sure you understand this entire program.

if ... else Statement

```
if (boolean expression)
{
   statements;
}
else
{
   statements;
}
```

In this case, if the *boolean expression* is true, Java will execute the *statements* directly after the if. If the *boolean expression* is false, it will only execute the *statements* after the else.

Extended if ... else Statement

You can chain many if ... else statements together. For example,

```
String grade = sc.next();  //read user input
if (grade.equals("A"))
      System.out.println("Very good!");
else if (grade.equals("B"))
      System.out.println("Good");
else if (grade.equals("C"))
      System.out.println("Poor");
else if (grade.equals("F"))
      System.out.println("Brutal");
else
      System.out.println("Invalid Grade");
System.out.println("Thanks for using my program").
```

If any of A, B, C, or F are entered, an appropriate message will be written, and then it will immediately print the "Thanks for using my program" message. If any other string is entered, the final else clause is invoked, and the "Invalid Grade" message is printed, followed by "Thanks for using my program".

Note the way we compare Strings using the .equals and the fact that if you have only 1 statement for each if or else if, you don't need the { }.

Compound Boolean Expressions

And, Or, Not

In Java, just like in Scratch, you can combine boolean statements using boolean operators:

Java	What it Means	Example
&&	And	`x > 3 && y > 4`
||	Or	`age > 16 || height <= 4`
!	Not	`if (!found)`

Examples

a&&b -- This statement is true only if both a and b are true. If ether a or b is false, the whole statement is false

a||b -- This statement is true if either a or b is true. It is only false if both a and b are false.

!a -- This statement reverses the truth of a. If a was false, the statement is true. If a was true, the statement would be false.

Exercises

Exercise 1: class Secret

Write a program that keeps a secret unless the correct password "dragon" is given.

Example Output 1

```
What's the password? dragon
Correct! My secret is: I am Batman.
```

Example Output 2

```
What's the password? Schlumbumb
Incorrect Password! My secret is still safe ...
```

Exercise 2: class GradeBot

Write a program that will help the user calculate their grade and percent on a test.

1. Ask the user what the test was out of
2. Ask the user for their test mark
3. Calculate and print their percentage grade as an integer (ie: 87%).
4. Figure out what letter grade that mark is, and write a message to the user to congratulate them on a job well done or encourage them to do better next time.
5. Find a way to prevent the user from crashing the program if they make the test out of zero, and deal with negative percentages and percentages greater than 100.

100-86: A 85-73: B 72-67: C+ 66-60: C 59-50: C- 49-0: F

Example Output 1:

```
What was your test out of? 100
What did you get? 99
You got 99%! That's an A ... keep up the good work.
```

Example Output 2:

```
What was your test out of? 25
What did you get? 15
You got 60%! That's a C ... you passed, but you can do better!
```

Example Ouput 3:

```
What was your test out of? 0
What did you get? 99
Silly human, that's not possible. Quit joking around.
```

Example Ouput 4:

```
What was your test out of? 100
What did you get? -4
Silly human, that's not possible. Quit joking around.
```

Exercise 3: class Magic8Ball

Using Math.Random, simulate a Magic 8 Ball. The user should be able to ask a yes or no question, and the Magic 8 Ball will return a random vague answer. You should have 20 possible answers that can be returned.

- Here is an example of a Magic 8 Ball simulator.
- Here is a list of the official Magic 8 Ball answers (although you are welcome to make up your own).