

Logic-1: squirrelPlay

The squirrels in Palo Alto spend most of the day playing. In particular, they play if the temperature is between 60 and 90 (inclusive). Unless it is summer, then the upper limit is 100 instead of 90. Given an int temperature and a boolean isSummer, return true if the squirrels play and false otherwise.

```
public boolean squirrelPlay(int temp, boolean isSummer) {  
  
}
```

Step 1

As usual, declare a variable of the same type as the return type of the method. Choose a variable name that makes sense in terms of what question the method is answering. Typically, initialize the variable to a negative/no answer – in this case, when the squirrels are NOT playing.

```
public boolean squirrelPlay(int temp, boolean isSummer) {  
    boolean playing = false;  
    return playing;  
}
```

Step 2

Add an if statement,

In the body of the if statement, with the specified condition (*they play if the temperature is between 60 and 90*). Set the return variable to the "positive" value when this condition will be true, in this case, when the squirrels play.

```
public boolean squirrelPlay(int temp, boolean isSummer) {  
    boolean playing = false;  
    if (60 <= temp && temp <= 90) {  
        playing = true;  
    }  
    return playing;  
}
```

Step 3

Notice that this condition is only for when it is **NOT** the summer (*Unless it is summer, then the upper limit is 100 instead of 90*). NEST the if statement that examines **temp** within an OUTER if statement that tests whether it's NOT the summer.

```
public boolean squirrelPlay(int temp, boolean isSummer) {  
    boolean playing = false;  
    if (!isSummer) {  
        if (60 <= temp && temp <= 90) {  
            playing = true;  
        }  
    }  
    return playing;  
}
```

Step 5

Duplicate the NESTED if statement. Change the condition in the second NESTED if statement to its opposite: IS the summer (**isSummer**). Change 90 to 100.

```
public boolean squirrelPlay(int temp, boolean isSummer) {
    boolean playing = false;
    if (!isSummer) {
        if (60 <= temp && temp <= 90) {
            playing = true;
        }
    }
    if (isSummer) {
        if (60 <= temp && temp <= 100) {
            playing = true;
        }
    }
    return playing;
}
```

Step 8

Notice that the ONLY difference between the 2 inner if statements is the upper temperatures: 90 and 100. Otherwise, the inner 2 if statements are identical.

We can actually simplify the program if the 2 outer if statements change only the upper limit temperature. We do this by declaring and initializing a new variable **upper**.

```
public boolean squirrelPlay(int temp, boolean isSummer) {
    boolean playing = false;
    int upper = 90;
    if (isSummer) {
        upper = 100;
    }
    if (!isSummer) {
        if (60 <= temp && temp <= 90) {
            playing = true;
        }
    }
    if (isSummer) {
        if (60 <= temp && temp <= 100) {
            playing = true;
        }
    }
    return playing;
}
```

Step 9

Because `isSummer` is now changing only what's absolutely necessary, we no longer need two NESTED if statements. So the lines highlighted in yellow can now be deleted, being careful to keep the 3 lines in red.

```
public boolean squirrelPlay(int temp, boolean isSummer) {
    boolean playing = false;
    int upper = 90;
    if (isSummer) {
        upper = 100;
    }
    if (!isSummer) {
        if (60 <= temp && temp <= 90) {
            playing = true;
        }
    }
    if (isSummer) {
        if (60 <= temp && temp <= 100) {
            playing = true;
        }
    }
    return playing;
}
```

Step 10

All that remains is to substitute `upper` for 90.
DONE!

```
public boolean squirrelPlay(int temp, boolean isSummer) {
    boolean playing = false;
    int upper = 90;
    if (isSummer) {
        upper = 100;
    }
    if (60 <= temp && temp <= upper) {
        playing = true;
    }
    return playing;
}
```