

## Logic-1: caughtSpeeding

You are driving a little too fast, and a police officer stops you. Write code to compute the result, encoded as an int value: 0=no ticket, 1=small ticket, 2=big ticket. If speed is 60 or less, the result is 0. If speed is between 61 and 80 inclusive, the result is 1. If speed is 81 or more, the result is 2. Unless it is your birthday -- on that day, your speed can be 5 higher in all cases.

```
public int caughtSpeeding(int speed, boolean isBirthday) {  
  
}
```

### 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, 0 represents no ticket.

```
public int caughtSpeeding(int speed, boolean isBirthday) {  
    int ticket = 0;  
    return ticket;  
}
```

### Step 2

Add if statements that encapsulate the 3 ranges of speeds and the type of ticket. *(If speed is 60 or less, the result is 0. If speed is between 61 and 80 inclusive, the result is 1. If speed is 81 or more, the result is 2.)*

```
public int caughtSpeeding(int speed, boolean isBirthday) {  
    int ticket = 0;  
    if (speed <= 60) {  
        ticket = 0;  
    }  
    if (61 <= speed && speed <= 80) {  
        ticket = 1;  
    }  
    if (81 <= speed) {  
        ticket = 2;  
    }  
    return ticket;  
}
```

### Step 3

Notice that the first condition sets the ticket to 0, which is the default value. We can therefore remove it, and add a comment that the default value represents speeds of 60 or less.

```
public int caughtSpeeding(int speed, boolean isBirthday) {  
    int ticket = 0; // default: speed <= 60  
    if (speed <= 60) {  
        ticket = 0;  
    }  
    if (61 <= speed && speed <= 80) {  
        ticket = 1;  
    }  
    if (81 <= speed) {  
        ticket = 2;  
    }  
    return ticket;  
}
```

#### Step 4

As usual, we can now NEST the two remaining if statements within two OUTER if statements that govern whether it is the driver's birthday or not. We can also increase by 5 each of the speeds of the inner if statements in the **isBirthday** branch. The program now works for all cases.

```
public int caughtSpeeding(int speed, boolean isBirthday) {
    int ticket = 0;
    if (!isBirthday) {
        if (61 <= speed && speed <= 80) {
            ticket = 1;
        }
        if (81 <= speed) {
            ticket = 2;
        }
    }
    if (isBirthday) {
        if (66 <= speed && speed <= 85) {
            ticket = 1;
        }
        if (86 <= speed) {
            ticket = 2;
        }
    }
    return ticket;
}
```

#### Step 5

However, look what happens if we rewrite those changed speeds, such as **66**, as **61 + 5** (and let the computer do the calculation!). We can now see that the only difference between the two OUTER if statement branches is the expression **+ 5**.

As with 03 squirrelPlay(), we can create a variable that will let us use one set of if statements, rather than two.

```
public int caughtSpeeding(int speed, boolean isBirthday) {
    int ticket = 0;
    if (!isBirthday) {
        if (61 <= speed && speed <= 80) {
            ticket = 1;
        }
        if (81 <= speed) {
            ticket = 2;
        }
    }
    if (isBirthday) {
        if (61 + 5 <= speed && speed <= 80 + 5) {
            ticket = 1;
        }
        if (81 + 5 <= speed) {
            ticket = 2;
        }
    }
    return ticket;
}
```

## Step 6

So ... we will create a variable `bDay` that will be assigned either the value 5 or 0, depending upon whether it's the driver's birthday or not.

```
public int caughtSpeeding(int speed, boolean isBirthday) {
    int ticket = 0;
    int bDay = 0;
    if (isBirthday) {
        bDay = 5;
    }
    if (!isBirthday) {
        if (61 <= speed && speed <= 80) {
            ticket = 1;
        }
        if (81 <= speed) {
            ticket = 2;
        }
    }
    if (isBirthday) {
        if (61 + 5 <= speed && speed <= 80 + 5) {
            ticket = 1;
        }
        if (81 + 5 <= speed) {
            ticket = 2;
        }
    }
    return ticket;
}
```

## Step 7

We can now replace all instances of `+ 5` with `+ bDay`, after which we can remove the OUTER if statements and the first branch.

```
public int caughtSpeeding(int speed, boolean isBirthday) {
    int ticket = 0;
    int bDay = 0;
    if (isBirthday) {
        bDay = 5;
    }
    if (!isBirthday) {
        if (61 <= speed && speed <= 80) {
            ticket = 1;
        }
        if (81 <= speed) {
            ticket = 2;
        }
    }
    if (isBirthday) {
        if (61 + bDay <= speed && speed <= 80 + bDay) {
            ticket = 1;
        }
        if (81 + bDay <= speed) {
            ticket = 2;
        }
    }
    return ticket;
}
```

**Step 8**  
DONE!

```
public int caughtSpeeding(int speed, boolean isBirthday) {
    int ticket = 0;
    int bDay = 0;
    if (isBirthday) {
        bDay = 5;
    }
    if (61 + bDay <= speed && speed <= 80 + bDay) {
        ticket = 1;
    }
    if (81 + bDay <= speed) {
        ticket = 2;
    }
    return ticket;
}
```