Logic-1: love6

The number 6 is a truly great number. Given two int values, a and b, return true if either one is 6. Or if their sum or difference is 6. Note: the function Math.abs(num) computes the absolute value of a number.

```
public boolean love6(int a, int b) {
}
```

Step 1
As usual, declare a variable of the same type as the return type of the method.
Call the variable **love**.
Initialize **love** to false.

```
public boolean love6(int a, int b) {
  boolean love = false;
  return love;
}
```

Step 2
Implement the first 2 conditions that will return true (*Given two int values, a and b, return true if either one is 6*).

```
public boolean love6(int a, int b) {
  boolean love = false;
  if (a == 6) {
    love = true;
  }
  if (b == 6) {
    love = true;
  }
  return love;
}
```

Step 3
Implement the next condition: *Or if their **sum** or difference is 6*.

```
public boolean love6(int a, int b) {
  boolean love = false;
  if (a == 6) {
    love = true;
  }
  if (b == 6) {
    love = true;
  }
  if (a + b == 6) {
    love = true;
  }
  return love;
}
```

Step 4
Implement the last condition: *Or if their sum or **difference** is 6*. Notice that the difference can be calculated in two ways: a – b or b – a. So to cover all possible cases, implement both.

```
public boolean love6(int a, int b) {
  boolean love = false;
  if (a == 6) {
    love = true;
  }
  if (b == 6) {
    love = true;
  }
  if (a + b == 6) {
    love = true;
  }
  if (a - b == 6) {
    love = true;
  }
  if (b - a == 6) {
    love = true;
  }
  return love;
}
```

Step 5
Although technically we are done, we can still make the code both more efficient and easier to read. The first change is to declare a variable **sum** and use that in place of a + b.

```
public boolean love6(int a, int b) {
  boolean love = false;
  int sum = a + b;
  if (a == 6) {
    love = true;
  }
  if (b == 6) {
    love = true;
  }
  if (sum == 6) {
    love = true;
  }
  if (a - b == 6) {
    love = true;
  }
  if (b - a == 6) {
    love = true;
  }
  return love;
}
```

Step 6
The second change is to declare 2 variables **diffAB** and **diffBA** and use them in place of a – b AND b - a.

```
public boolean love6(int a, int b) {
  boolean love = false;
  int sum = a + b;
  int diffAB = a - b;
  int diffBA = b - a;
  if (a == 6) {
    love = true;
  }
  if (b == 6) {
    love = true;
  }
  if (sum == 6) {
    love = true;
  }
  if (diffAB == 6) {
    love = true;
  }
  if (diffBA == 6) {
    love = true;
  }
  return love;
}
```

Step 7
Notice that we can collapse the 2 variables **diffAB** and **diffBA** into a single variable **diff** if we take the **ABSOLUTE VALUE** of the difference. Java has a method called **Math.abs()** that does exactly this.

```
public boolean love6(int a, int b) {
  boolean love = false;
  int sum = a + b;
  int diff = Math.abs(a - b);
  if (a == 6) {
    love = true;
  }
  if (b == 6) {
    love = true;
  }
  if (sum == 6) {
    love = true;
  }
  if (diff == 6) {
    love = true;
  }
  return love;
}
```

Step 8
Finally we can collapse multiple sequential if statements into single if statements if we use the **boolean operator OR ||** .

```
public boolean love6(int a, int b) {
  boolean love = false;
  int sum = a + b;
  int diff = Math.abs(a - b);
  if (a == 6 || b == 6) {
    love = true;
  }
  if (sum == 6 || diff == 6) {
    love = true;
  }
  return love;
}
```

OR

```
public boolean love6(int a, int b) {
  boolean love = false;
  int sum = a + b;
  int diff = Math.abs(a - b);
  if (a == 6 || b == 6 || sum == 6 || diff == 6) {
    love = true;
  }
  return love;
}
```