

## (Bitwise Operations)

## العمليات المنطقية على مستوى البت (Bitwise Operations)

هي عمليات حسابية تتم على مستوى البت الواحد (٠ أو ١) داخل الأعداد الثنائية (binary numbers).  
بعبارة أخرى، هي عمليات تتم مباشرة على التمثيل الثنائي للأعداد، وليس على قيمها العشرية

## الاستخدامات :

- التلاعب بالبيانات: تستخدم لتغيير قيم بتات معينة داخل عدد ثنائي، مما يتيح لنا التحكم الدقيق في البيانات.
- التشفير: تستخدم في العديد من خوارزميات التشفير لضمان أمان البيانات.
- التحكم في الأجهزة: تستخدم للتحكم في أجهزة الحاسوب على مستوى منخفض، مثل التحكم في مداخل ومخارج الجهاز.
- البرمجة على مستوى النظام: تستخدم في البرمجة على مستوى النظام (system programming) حيث نحتاج إلى التعامل مع البيانات على مستوى البت.

## أنواع العمليات المنطقية على مستوى البت:

### 1. العامل (~) (Bitwise Complement OR)

العامل ~ يقلب البتات التي تساوي ٠ إلى ١ و يقلب البتات التي تساوي ١ إلى ٠ .

بعدها يتم حساب الناتج باتتباع مبدأ اسمه single precision floating point number.

### مثال

JavaApplication41 - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Start Page JavaApplication40.java dog\_1.java Car.java JavaApplication41.java

Source History

```

package javaapplication41;
public class JavaApplication41 {

    public static void main(String[] args) {

        int a = 10;        // 10 = 000000000000000000000000000001010
        int c = ~a;        // c = 111111111111111111111111111110101 = -11

        System.out.print("~" + a + " = " + c);

    }
}

```

Output - JavaApplication41 (run)

```

run:
~10 = -11BUILD SUCCESSFUL (total time: 0 seconds)

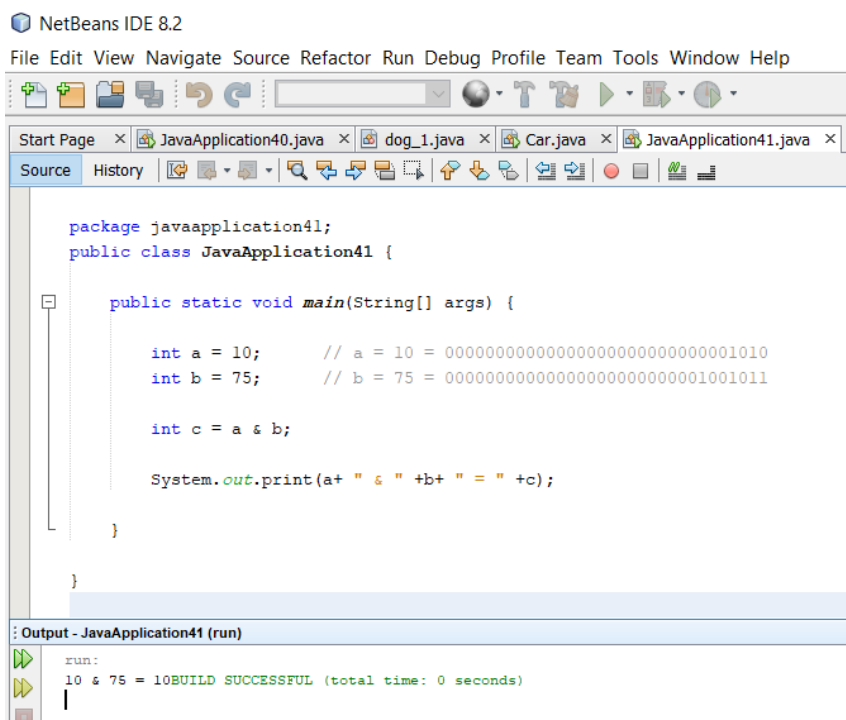
```

## 2. العامل & (Bitwise AND)

يحسب ناتج جمع البتات (Bits) المشتركة بين قيمتين.

يُستخدم لإجراء عملية AND على مستوى البت بين رقمين. البت الناتج يكون ١ فقط إذا كان البتان المتقابلان في الرقمين كلاهما ١.

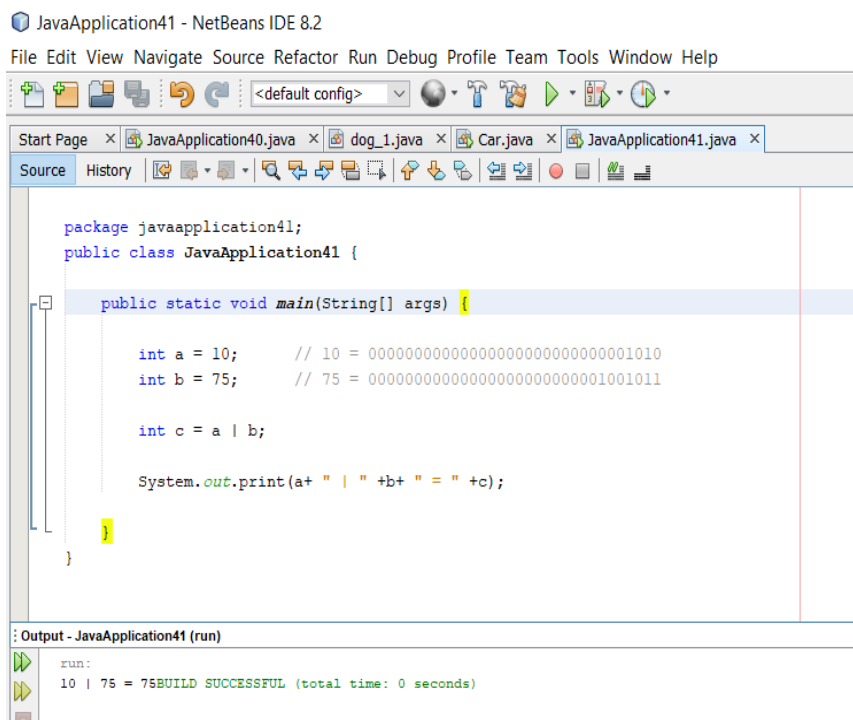
### مثال



### ٣. المشغل (Bitwise OR)

يُستخدم لإجراء عملية OR على مستوى البت بين رقمين. البت الناتج يكون ١ إذا كان أحد البتين أو كلاهما يساوي ١.

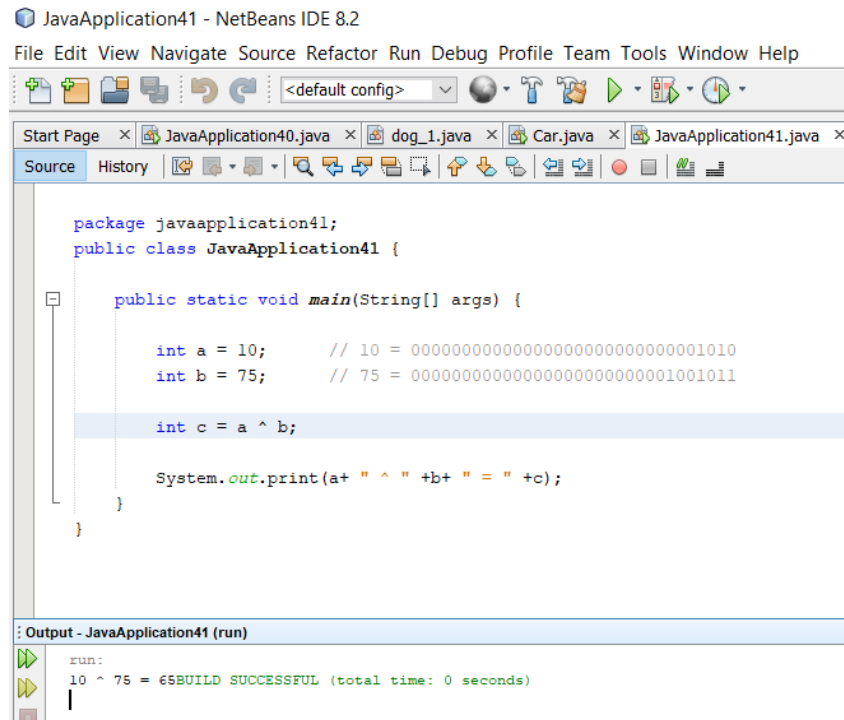
### مثال



#### 4. المشغل $\wedge$ (Bitwise XOR)

يُستخدم لإجراء عملية XOR على مستوى البت بين رقمين. البت الناتج يكون 1 إذا كان البتان المتقابلان مختلفين.

مثال



```
package javaapplication41;
public class JavaApplication41 {

    public static void main(String[] args) {

        int a = 10;        // 10 = 00000000000000000000000000001010
        int b = 75;        // 75 = 00000000000000000000000000001001011

        int c = a ^ b;

        System.out.print(a + " ^ " + b + " = " + c);

    }

}
```

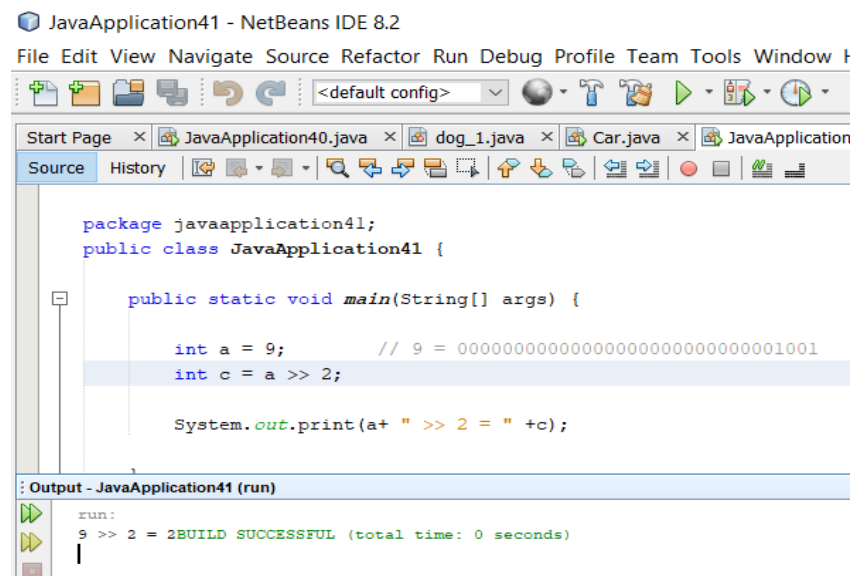
Output - JavaApplication41 (run)

```
run:
10 ^ 75 = 65BUILD SUCCESSFUL (total time: 0 seconds)
```

#### 5. المشغل $>>$ (Left Shift)

يُستخدم لتحريك البتات إلى اليسار بعدد معين من المرات، ويتم ملء الفراغات من اليمين بالأصفار. يُضاعف القيم (يشبه الضرب بالقوى الموجبة للعدد 2).

مثال



```
package javaapplication41;
public class JavaApplication41 {

    public static void main(String[] args) {

        int a = 9;        // 9 = 00000000000000000000000000001001
        int c = a >> 2;

        System.out.print(a + " >> 2 = " + c);

    }

}
```

Output - JavaApplication41 (run)

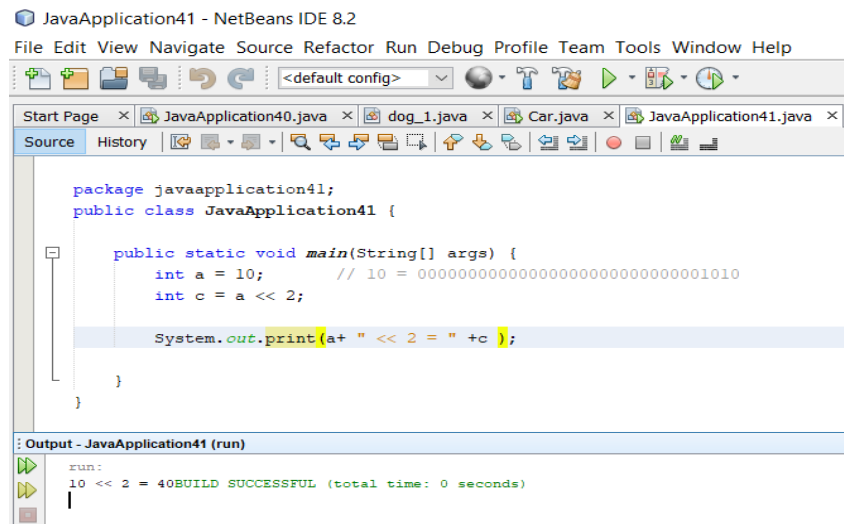
```
run:
9 >> 2 = 2BUILD SUCCESSFUL (total time: 0 seconds)
```

## 6. المشغل << (Right Shift with Sign Extension)

يُستخدم لتحريك البتات إلى اليمين بعدد معين من المرات، ويتم ملء الفراغات من اليسار باستخدام البت الخاص بالإشارة (٠ إن كان الرقم موجبًا، و ١ إن كان الرقم سالبًا).

يُستخدم لتقسيم القيم (يشبه القسمة على القوى الموجبة للعدد ٢).

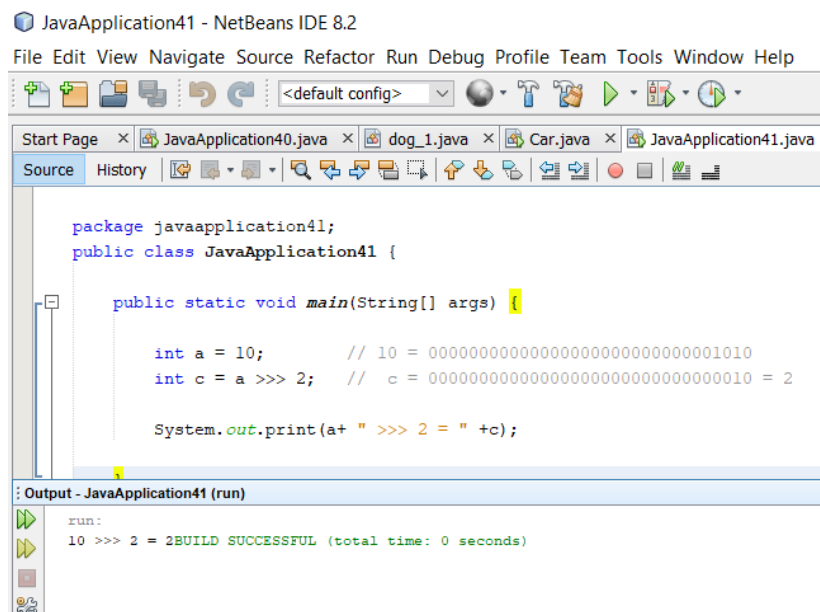
مثال



### 7. المشغل <<< (Right Shift with Zero Fill)

الوصف: يُستخدم لتحريك البتات إلى اليمين بعدد معين من المرات، ويتم ملء الفراغات من اليسار دائماً بالأصفار بغض النظر عن إشارة الرقم.

### مثال



**الطالبة: سلسبيل ماجد الوراق**

**التخصص:** نظم معلومات (مستوى ثانى)