

## Seçim 2

İşlem	Örnek
<code>if</code>	<code>if (x &gt; 0) { sout("Pozitif"); }</code>
<code>if ... else</code>	<code>if (x &gt; 0) { sout("Pozitif"); } else { sout("Pozitif değil"); }</code>
<code>if ... elseif ... else</code>	<code>if (x &gt; 0) { sout("Pozitif"); } else if (x &lt; 0) { sout("Negatif"); } else { sout("Sıfır"); }</code>
<code>switch</code>	<code>switch (x) { case 1: sout("Bir"); break; case 2: sout("İki"); break; default: sout("Diğer"); }</code>
İç İçe if ifadesi	<code>if (x &gt; 0) { if (x % 2 == 0) { sout("Pozitif ve Çift"); } }</code>
Koşullu Operatörler	<code>boolean isTrue = (x &gt; 0) &amp;&amp; (x &lt; 10);</code>

İşlem	Açıklama	Örnek	Çıktı Örneği
<code>+</code> (İkili toplama)	İkili aritmetik toplama.	<code>int a = 5 + 3;</code>	a, 8'e eşittir
<code>-</code> (İkili çıkarma)	İkili aritmetik çıkarma.	<code>int b = 8 - 4;</code>	b, 4'e eşittir
<code>*</code> (Çarpma)	İkili çarpma.	<code>int c = 6 * 4;</code>	c, 24'e eşittir
<code>/</code> (Bölme)	İkili bölme.	<code>int d = 20 / 5;</code>	d, 4'e eşittir
<code>%</code> (Modül)	İkili modül (bölümün kalanı).	<code>int e = 17 % 5;</code>	e, 2'ye eşittir
<code>+</code> (Ünary artı)	Ünary artı.	<code>int f = +5;</code>	f, 5'e eşittir
<code>-</code> (Ünary eksi)	Ünary eksi.	<code>int g = -7;</code>	g, -7'ye eşittir
<code>++var</code> (Ön-ekleme)	Ön-ekleme.	<code>int h = 5; ++h;</code>	h, 6'ya dönüşür
<code>--var</code> (Ön-azaltma)	Ön-azaltma.	<code>int i = 10; --i;</code>	i, 9'a dönüşür
<code>var++</code> (Son-ekleme)	Son-ekleme.	<code>int j = 5; j++;</code>	j, 6'ya dönüşür
<code>var--</code> (Son-azaltma)	Son-azaltma.	<code>int k = 10; k--;</code>	k, 9'a dönüşür
<code>!</code> (Değil)	Mantıksal DEĞİL.	<code>boolean l = !flag;</code>	l, false olur
<code>&lt;</code> (Küçük)	Küçüktür karşılaştırması.	<code>boolean m = (5 &lt; 10);</code>	m, true olur
<code>&gt;</code> (Büyük)	Büyüktür karşılaştırması.	<code>boolean n = (10 &gt; 5);</code>	n, true olur
<code>!=</code> (Eşit değil)	Eşit değil karşılaştırması.	<code>boolean o = (3 != 5);</code>	o, true olur
<code>==</code> (Eşit)	Eşit karşılaştırması.	<code>boolean p = (5 == 5);</code>	p, true olur
<code>&lt;=</code> (Küçük veya eşit)	Küçük veya eşit karşılaştırması.	<code>boolean q = (10 &lt;= 15);</code>	q, true olur
<code>&gt;=</code> (Büyük veya eşit)	Büyük veya eşit karşılaştırması.	<code>boolean r = (20 &gt;= 10);</code>	r, true olur
<code>&amp;&amp;</code> (Mantıksal VE)	Koşullu VE.	<code>boolean s = (true &amp;&amp; false);</code>	s, false olur
<code>  </code> (Mantıksal VEYA)	Koşullu VEYA.	<code>boolean t = (true    false);</code>	t, true olur
<code>^</code> (Mantıksal XOR)	Koşullu XOR (özel VEYA).	<code>boolean u = (true ^ false);</code>	u, true olur
<code>=</code> (Atama)	Atama.	<code>int w = 5;</code>	w, 5 olur

İşlem	Açıklama	Örnek	Çıktı Örneği
<b>+=</b> (Ekle ve ata)	Ekle ve ata.	<code>int x = 5; x += 3;</code>	x, 8 olur
<b>-=</b> (Çıkar ve ata)	Çıkar ve ata.	<code>int y = 10; y -= 4;</code>	y, 6 olur
<b>*=</b> (Çarp ve ata)	Çarp ve ata.	<code>int z = 20; z *= 2;</code>	z, 40 olur
<b>/=</b> (Böl ve ata)	Böl ve ata.	<code>int m = 15; m /= 3;</code>	m, 5 olur
<b>%=</b> (Modül ve ata)	Modül ve ata.	<code>int n = 12; n %= 5;</code>	n, 2 olur

1. Kodun Path A yazdırması için boşlukları doldurun.

```
class punchcard {  
    public static void main(String[] args) {  
        int x = 4;  
        ----- (x) {  
            ----- 4:  
                System.out.println("Path A");  
            -----;  
        default:  
            System.out.println("Path B");  
            break;  
        }  
    }  
}
```

- A) switch, case, break
  - B) if, else, continue
  - C) if, else, break
  - D) switch, case, continue
- 

2. Aşağıdaki kod parçası için çıktı ne olur?

```
class punchcard {  
    public static void main(String[] args) {  
        int y = 4;  
        switch (y) {  
            default:  
                System.out.println("Default");  
            case 1:  
                System.out.println("One");  
                break;  
            case 2:  
                System.out.println("Two");  
            case 3:  
                System.out.println("Three");  
                break;  
        }  
    }  
}
```

- A) Default
- B) One
- C) Default \n One

- D) Default \n One \n Two \n Three
  - E) Default \n Two \n Three
- 

3. Aşağıdaki Java kodunun çıktısı nedir?

```
class punchcard {  
    public static void main(String[] args) {  
        int z = 2;  
        switch (z) {  
            case 1:  
                System.out.println("Case 1");  
                break;  
            case 2:  
            case 3:  
                System.out.println("Case 2 or 3");  
                break;  
            default:  
                System.out.println("Default");  
        }  
    }  
}
```

- A) Case 1
  - B) Case 2 or 3
  - C) Default
  - D) Case 2 or 3 \n Default
  - E) Çıktı yok
  - F) Hata
- 

4. Aşağıdaki Java kodunun çıktısı nedir?

```
class punchcard {  
    public static void main(String[] args) {  
        int input = 7;  
        switch (input % 5) {  
            case 1:  
                System.out.println("One");  
                break;  
            case 2:  
                System.out.println("Two");  
                break;  
            case 3:
```

```

        System.out.println("Three");
        break;
    default:
        System.out.println("Default");
    }
}
}

```

- A) One
- B) Two
- C) Three
- D) Default

- 
- 5 ila 7. sorular ekstradır. İsterseniz bunları atlayabilirsiniz.
  - Bu switch özellikleri Java 14 veya daha üstünü gerektirir.

5. Aşağıdaki Java kodunun çıktısı nedir?

```

class punchcard {
    public static void main(String[] args) {
        int dayofweek = 6;
        switch (dayofweek) {
            case 1, 2, 3, 4, 5:
                System.out.println("bekle, bugün ders mi var?!");
                break;
            case 6, 7:
                System.out.println("partiye gidiyoruzzzzz");
                break;
            default:
                System.out.println("geçerli bir gün değil!");
        }
    }
}

```

- A) bekle, bugün ders mi var?!!
- B) partiye gidiyoruzzzzz
- C) Geçerli bir gün değil!
- D) Hata

---

6. Aşağıdaki Java kodunun çıktısı nedir?

```

[class] (class) punchcard {
    public static void main(String[] args) {

```

```

int dayOfWeek = 6;
switch (dayOfWeek) {
    case 1, 2, 3, 4, 5 -> System.out.println("bekle, bugün ders mi var?!");
    case 6, 7 -> System.out.println("partiye gidiyoruzzzzz");
    default -> System.out.println("Geçerli bir gün değil!");
}
}
}

```

- A) bekle, bugün ders mi var?!!
- B) partiye gidiyoruzzzzz
- C) Geçerli bir gün değil!
- D) Hata

---

7. Aşağıdaki Java kodunun çıktısı nedir?

```

class punchcard {
    public static void main(String[] args) {
        int dayOfWeek = 7;
        int day = 23, month = 5, year = 2021;

        String date = switch (dayOfWeek) {
            case 1 -> "Pazartesi";
            case 2 -> "Salı";
            case 3 -> "Çarşamba";
            case 4 -> "Perşembe";
            case 5 -> "Cuma";
            case 6 -> "Cumartesi";
            case 7 -> "Pazar";
            default -> "Geçersiz haftanın günü!";
        }

        + ", " + day + ". "
        + switch (month) {
            case 1 -> "Ocak";
            case 2 -> "Şubat";
            case 3 -> "Mart";
            case 4 -> "Nisan";
            case 5 -> "Mayıs";
            case 6 -> "Haziran";
            case 7 -> "Temmuz";
            case 8 -> "Ağustos";
            case 9 -> "Eylül";
            case 10 -> "Ekim";
        }
    }
}

```

```

        case 11 -> "Kasım";
        case 12 -> "Aralık";
        default -> "Geçersiz ay!";
    }
    + " " + year;
    System.out.println(date + "\n");
}
}

```

- A) Pazar, 23. Mayıs 2021
- B) Geçersiz haftanın günü!, 23. Geçersiz ay! 2021
- C) Geçersiz haftanın günü!, 23. Mayıs 2021
- D) Perşembe, 23. Nisan 2021
- E) Hata

8. Renk eşleştirme oyununu simüle eden bir program oluşturuyorsunuz. Her rengin belirli bir sayıya karşılık geldiği bir renk tablosu bulunmaktadır. Göreviniz, kullanıcıdan string türünde (`color`) bir giriş almak olan bir kod yazmaktır. Giriş rengi, aşağıdaki renk tablosuna dayalı olarak bir sayısal koda eşleştirilecektir:

Renk kodları şöyledir: - Kırmızı - 101 - Mavi - 202 - Yeşil - 303 - Sarı - 404 - Turuncu - 505

Ardından, karşılık gelen sayısal kodu konsola yazdırın.

Örnek fonksiyon imzası:

```

public class Main {
    public static void main(String[] args) {
        // bir tarama nesnesi oluşturun

        // kullanıcıdan rengi alın

        // switch-case kullanarak karşılık gelen sayısal kodu yazdırmak için if-else veya switch-case kullanın
    }
}

```

#### Kısıtlamalar:

- Giriş `color` her zaman “Kırmızı”, “Mavi”, “Yeşil”, “Sarı”, “Turuncu” arasında geçerli bir dize olacaktır.

#### Örnek:

Bu soru, adayların `switch-case` ifadesini kullanarak giriş renk dizesini karşılık gelen sayısal koda verimli ve yaratıcı bir şekilde eşleştirmesini istemektedir. Hedef, `switch-case` yapısını kullanarak kompakt ve zarif bir `getColorCode` fonksiyonu uygulamaktır.

Cevaplar:

1. A
2. C
3. B
4. B
5. B
6. B
7. A
8. Olası bir çözüm:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String color = scanner.nextLine();
        switch (color) {
            case "Kırmızı":
                System.out.println("101");
                break;
            case "Mavi":
                System.out.println("202");
                break;
            case "Yeşil":
                System.out.println("303");
                break;
            case "Sarı":
                System.out.println("404");
                break;
            case "Turuncu":
                System.out.println("505");
                break;
            default:
                System.out.println("Geçersiz renk");
        }
    }
}
```