

```

public class Card implements Comparable {

    private int value;

    /**
     * Constructs a new Card object from a value
     */
    public Card(int inValue) {
        value = inValue;
    }

    /**
     * Constructs a new Card object from a description
     * @param inDesc the desc of the card (e.g. A, 2..9, T, J, Q, K)
     */
    public Card(String inDesc) {

        if(inDesc.equals("A"))
            value = 1;
        else if(inDesc.equals("T"))
            value = 10;
        else if(inDesc.equals("J"))
            value = 11;
        else if(inDesc.equals("Q"))
            value = 12;
        else if(inDesc.equals("K"))
            value = 13;
        else
            value = Integer.parseInt(inDesc);
    }
}

```

```
}
```

```
/**
```

```
 * Returns a string representation of the card.
```

```
*/
```

```
public String toString() {
```

```
    if(value == 1)
```

```
        return "A";
```

```
    else if (value == 10)
```

```
        return "T";
```

```
    else if (value == 11)
```

```
        return "J";
```

```
    else if (value == 12)
```

```
        return "Q";
```

```
    else if (value == 13)
```

```
        return "K";
```

```
    else
```

```
        return String.valueOf(value);
```

```
}
```

```
/**
```

```
 * Compares two cards
```

```
*/
```

```
public int compareTo(Object o) {
```

```
    Card c = (Card) o;
```

```
    if (value == c.value)
```

```
        return 0;
```

```
    else if (value > c.value)
```

```
        return 1;
```

```
        else
            return -1;
    }

    /**
     * Returns true if this card equals a given card
     */
    public boolean equals(Object o) {
        return compareTo(o) == 0;
    }

    /**
     * Returns the matching card to this card
     */
    public Card getMatchingCard() {
        return new Card(11 - value);
    }
}
```