Object Oriented Programming using Java

Prepared By:
Suyel, PhD
Assistant Professor
Dept. of CSE, NIT Patna

Outline

1. Conditional Statement

Conditional Statement

- A conditional statement allows us to choose which statement will be executed. So, it is sometimes called as selection statement.
- Conditional statement gives us the power to make decision.
- Java has the following conditional statements:
  - if
  - if-else
  - if-else-if
  - Nested if
  - Switch case

if Statement

- if statement is the most simple decision making statement.
- It is used to decide whether a certain statement or block of statements would be executed.
- There can be multiple if statements in a java program. In this case, the statements would be executed, if the corresponding conditions are true.
- Syntax:

```
 if(condition)
{
   Statement 1;
   Statement 2;
}
```

if Statement (Cont…)

- if is a reserved word in java.
- The condition must be a Boolean expression. It must evaluate to either true or false.
- If we do not give the curly braces ‘{’ and ‘}’ after “if(condition)”, then, by default if statement considers the immediate statement inside its block. Example:

```
if(condition)
   statement 1;                //Statement 1 will be executed
   statement 2;
```

if Statement (Cont…)

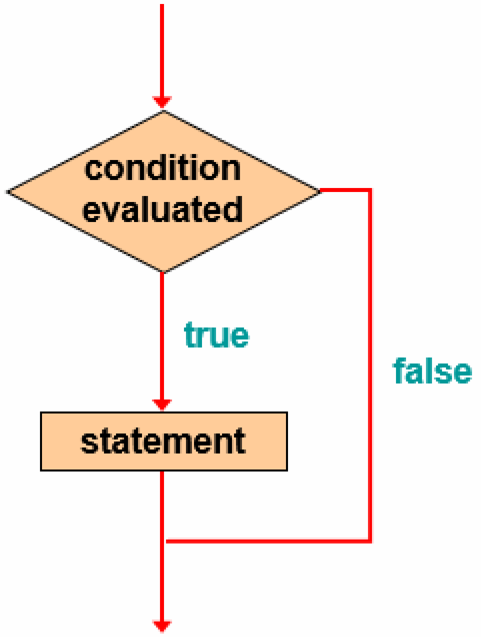
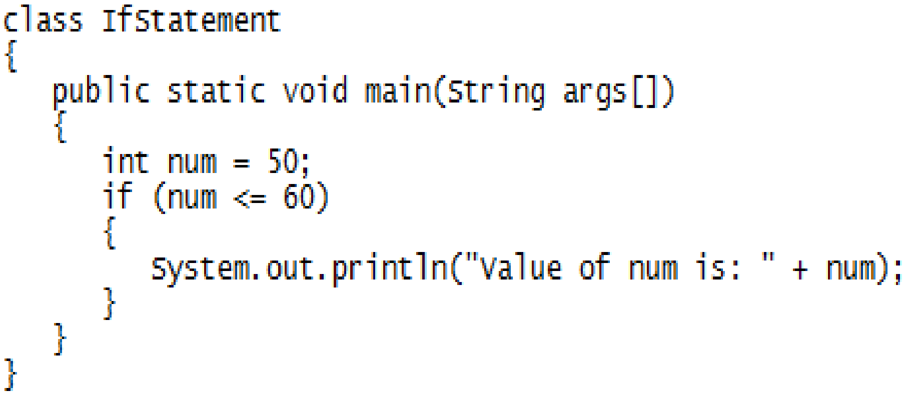


Fig. 1: Flowchart of if statement

if Statement (Cont…)



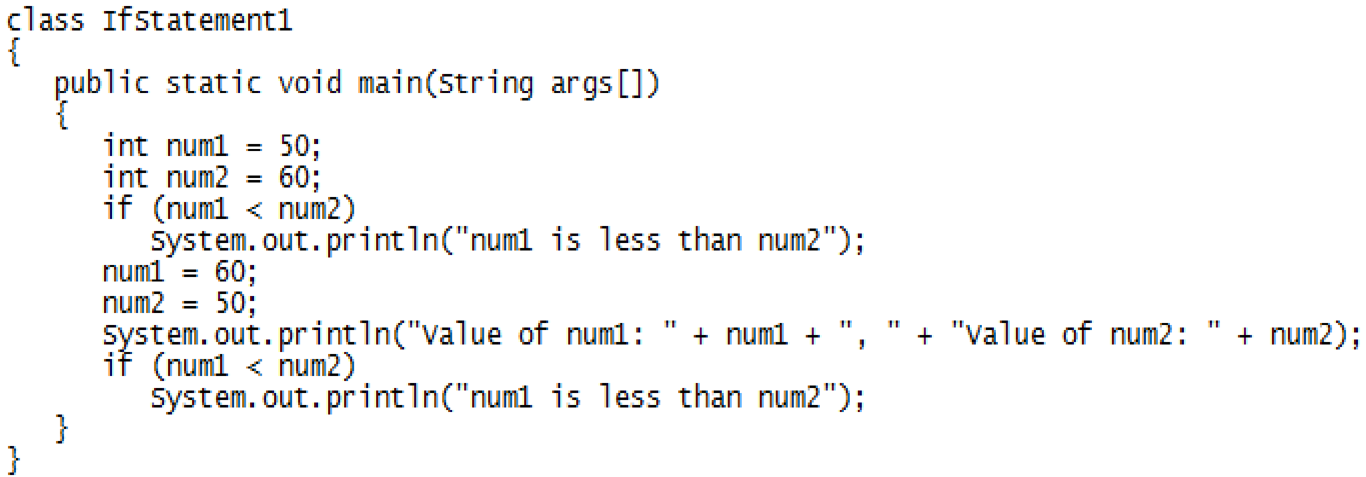
```
class IfStatement
{
   public static void main(String args[])
   {
      int num = 50;
      if (num <= 60)
      {
         System.out.println("Value of num is: " + num);
      }
   }
}
```

if Statement (Cont…)

- Output

```
Value of num is: 50
```

if Statement (Cont…)



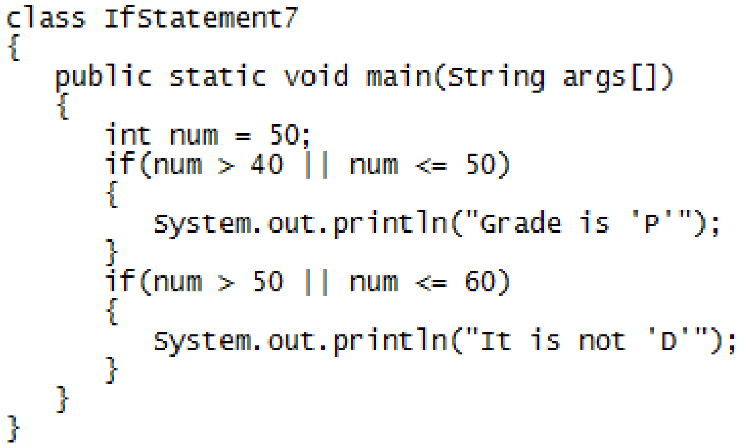
```
class IfStatement1
{
   public static void main(String args[])
   {
      int num1 = 50;
      int num2 = 60;
      if (num1 < num2)
         System.out.println("num1 is less than num2");
      num1 = 60;
      num2 = 50;
      System.out.println("Value of num1: " + num1 + ", " + "Value of num2: " + num2);
      if (num1 < num2)
         System.out.println("num1 is less than num2");
   }
}
```

if Statement (Cont…)

- Output

```
num1 is less than num2
Value of num1: 60, Value of num2: 50
```

if Statement (Cont…)



```
class IfStatement7
{
    public static void main(String args[])
    {
        int num = 50;
        if(num > 40 || num <= 50)
        {
            System.out.println("Grade is 'P'");
        }
        if(num > 50 || num <= 60)
        {
            System.out.println("It is not 'D'");
        }
    }
}
```

if Statement (Cont…)

- Output

```
Grade is ‘P’
It is not ‘D’
```

if-else Statement

- We use the else statement to specify a block of code to be executed, if the condition is false.
- Syntax:

```
if(condition)
{
   Statement 1;            //Block of code to be executed, if the condition is true
}
else
{
   Statement 2;            //Block of code to be executed, if the condition is false
}
```

if-else Statement (Cont…)

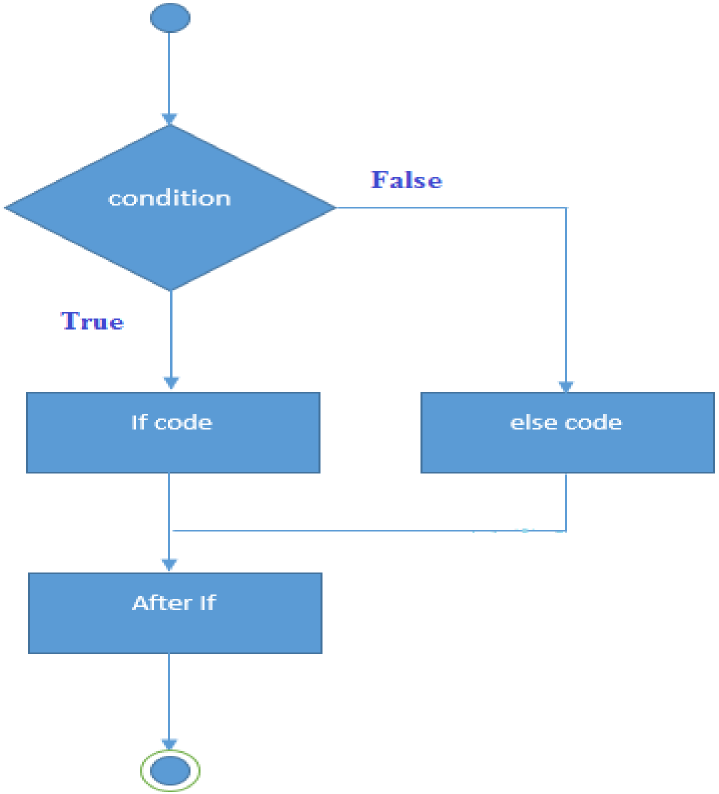
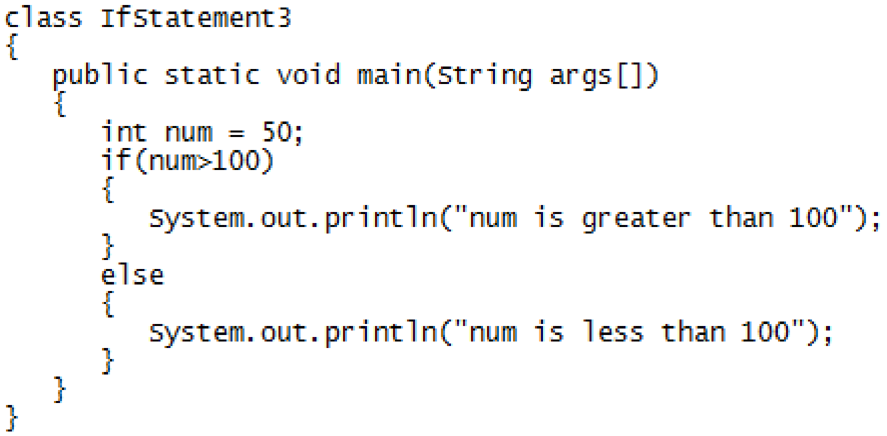


Fig. 2: Flowchart of if-else statement

if-else Statement (Cont…)



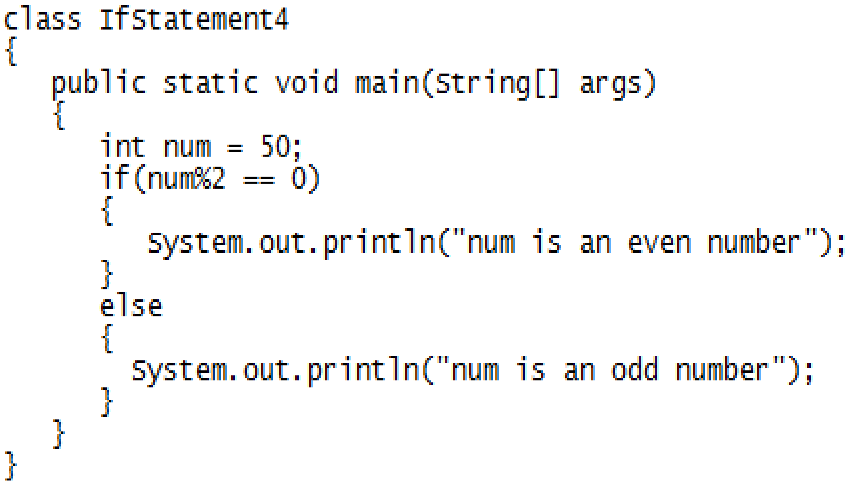
```
class IfStatement3
{
   public static void main(String args[])
   {
      int num = 50;
      if(num>100)
      {
         System.out.println("num is greater than 100");
      }
      else
      {
         System.out.println("num is less than 100");
      }
   }
}
```

if-else Statement (Cont…)

- Output

```
num is less than 100
```

if-else Statement (Cont…)



```
class IfStatement4
{
   public static void main(String[] args)
   {
      int num = 50;
      if(num%2 == 0)
      {
         System.out.println("num is an even number");
      }
      else
      {
        System.out.println("num is an odd number");
      }
   }
}
```

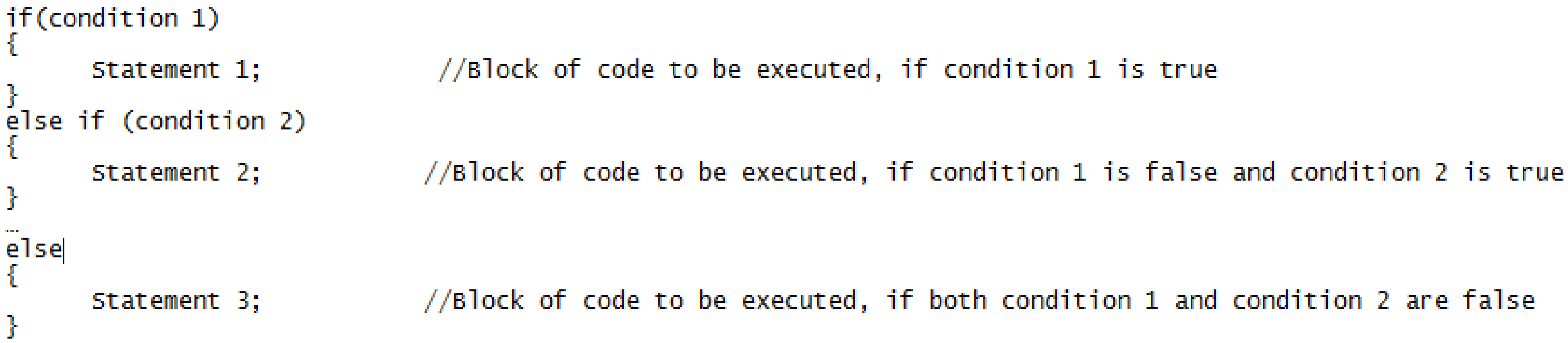
if-else Statement (Cont…)

- Output

```
num is an even number
```

if-else-if Statement

- if-else-if statement is used, when we need to check multiple conditions.
- We can have only one if statement and one else statement in this case. However, we can have multiple else-if statement.
- Here, as soon as the condition is met, the corresponding set of statements get executed and rest gets ignored. Syntax:



```
if(condition 1)
{
        Statement 1;              //Block of code to be executed, if condition 1 is true
}
else if (condition 2)
{
        Statement 2;             //Block of code to be executed, if condition 1 is false and condition 2 is true
}
…
else
{
        Statement 3;             //Block of code to be executed, if both condition 1 and condition 2 are false
}
```

if-else-if Statement (Cont…)

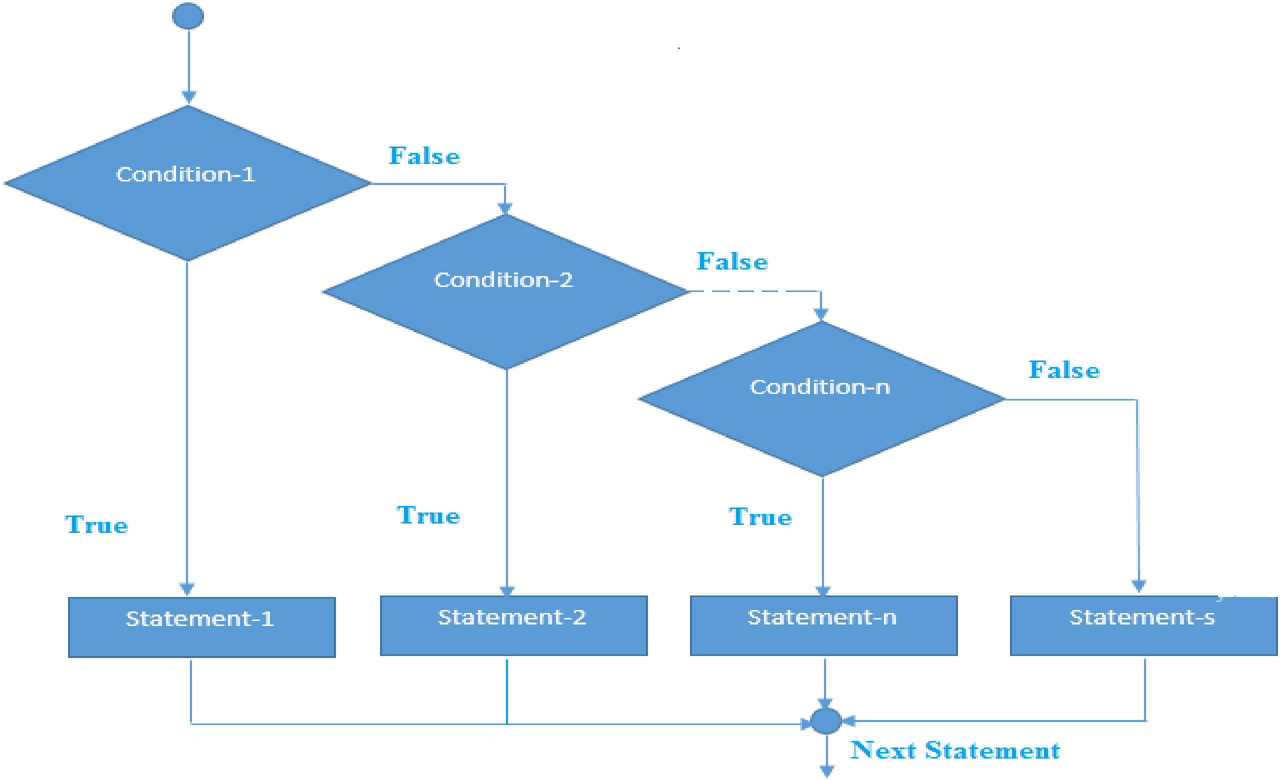
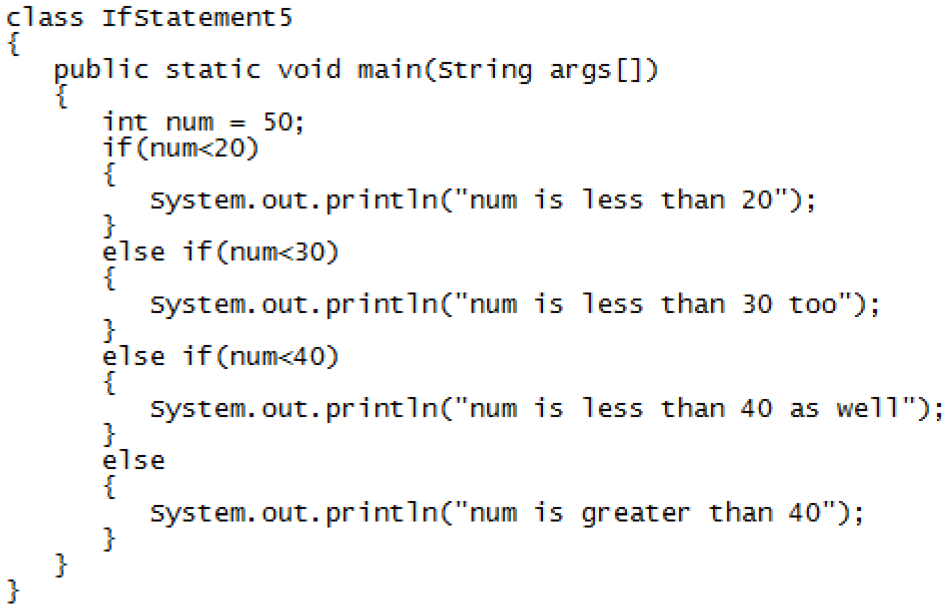


Fig. 3: Flowchart if-else-if statement

if-else-if Statement (Cont…)



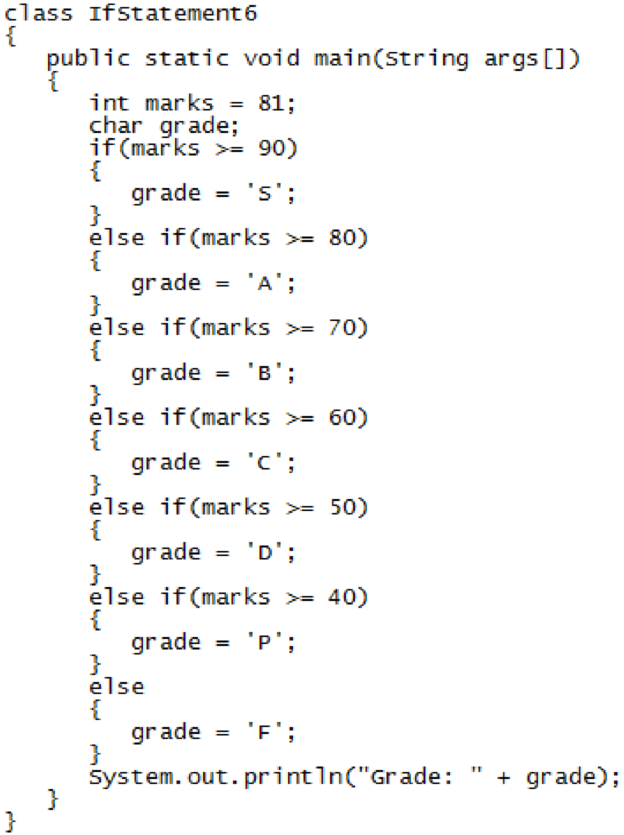
```java
class IfStatement5
{
    public static void main(String args[])
    {
        int num = 50;
        if(num<20)
        {
            System.out.println("num is less than 20");
        }
        else if(num<30)
        {
            System.out.println("num is less than 30 too");
        }
        else if(num<40)
        {
            System.out.println("num is less than 40 as well");
        }
        else
        {
            System.out.println("num is greater than 40");
        }
    }
}
```

if-else-if Statement (Cont…)

- Output

```
num is greater than 40
```

if-else-if Statement (Cont…)



```
class IfStatement6
{
    public static void main(String args[])
    {
        int marks = 81;
        char grade;
        if(marks >= 90)
        {
            grade = 'S';
        }
        else if(marks >= 80)
        {
            grade = 'A';
        }
        else if(marks >= 70)
        {
            grade = 'B';
        }
        else if(marks >= 60)
        {
            grade = 'C';
        }
        else if(marks >= 50)
        {
            grade = 'D';
        }
        else if(marks >= 40)
        {
            grade = 'P';
        }
        else
        {
            grade = 'F';
        }
        System.out.println("Grade: " + grade);
    }
}
```

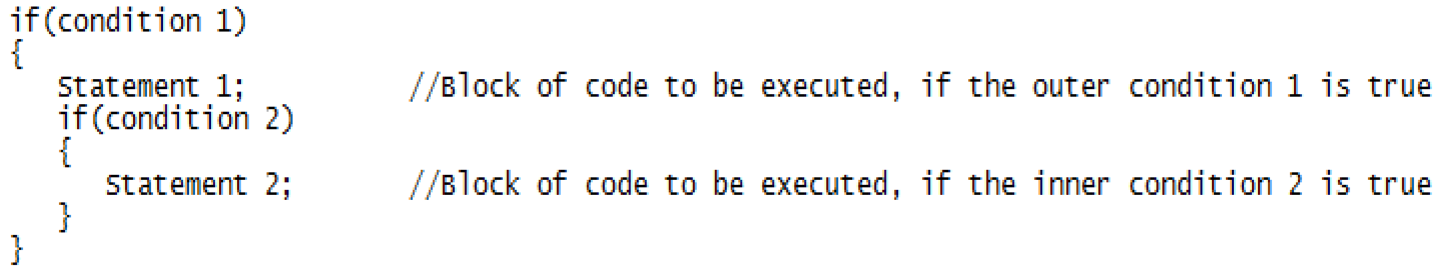
if-else-if Statement (Cont…)

- Output

```
Grade: A
```

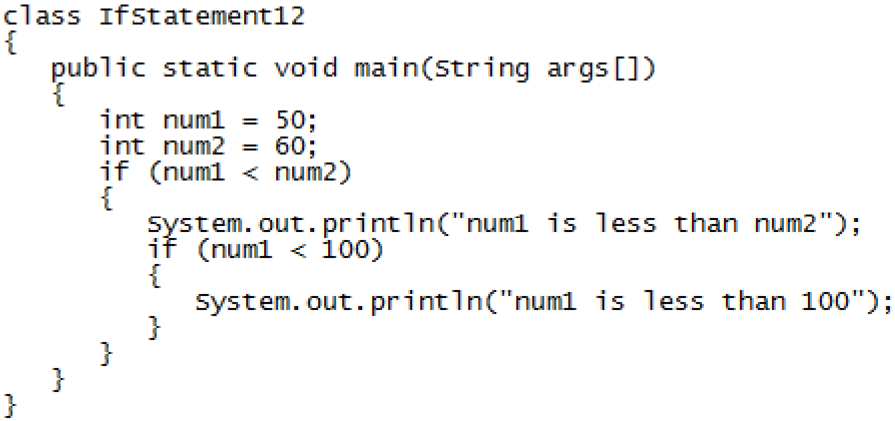
Nested if Statement

- It is an if statement inside another if statement.
- If the outer if condition is true, the section of code under outer if condition is executed.
- If inner if condition is true, the section of code under inner if condition is executed.
- Syntax:



```
if(condition 1)
{
   Statement 1;          //Block of code to be executed, if the outer condition 1 is true
   if(condition 2)
   {
      Statement 2;       //Block of code to be executed, if the inner condition 2 is true
   }
}
```

Nested if Statement (Cont…)



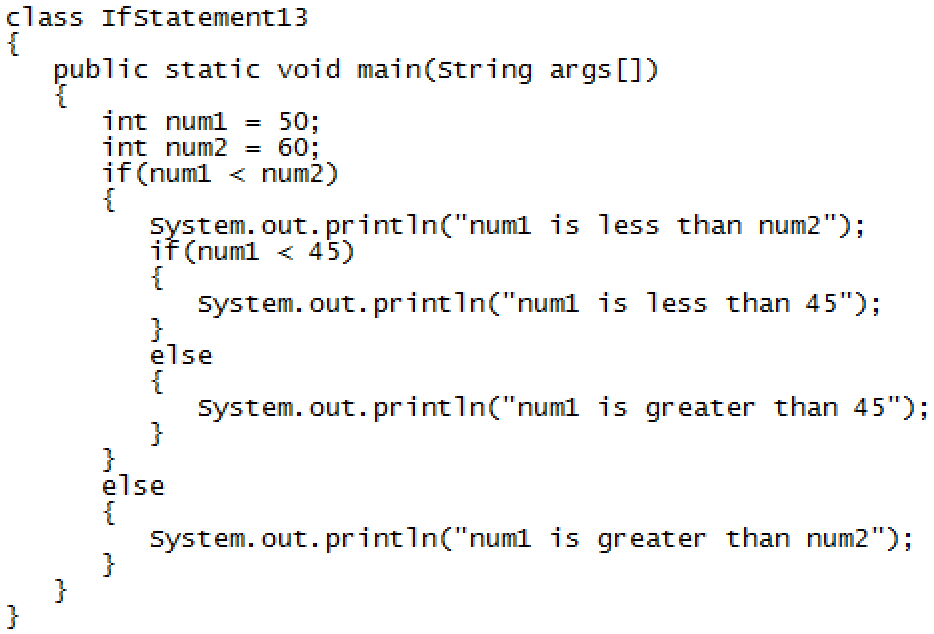
```
class IfStatement12
{
    public static void main(String args[])
    {
        int num1 = 50;
        int num2 = 60;
        if (num1 < num2)
        {
            System.out.println("num1 is less than num2");
            if (num1 < 100)
            {
                System.out.println("num1 is less than 100");
            }
        }
    }
}
```

Nested if Statement (Cont…)

- Output

```
num1 is less than num2
num1 is less than 100
```

Nested if Statement (Cont…)



```
class IfStatement13
{
   public static void main(String args[])
   {
      int num1 = 50;
      int num2 = 60;
      if(num1 < num2)
      {
         System.out.println("num1 is less than num2");
         if(num1 < 45)
         {
            System.out.println("num1 is less than 45");
         }
         else
         {
            System.out.println("num1 is greater than 45");
         }
      }
      else
      {
         System.out.println("num1 is greater than num2");
      }
   }
}
```

Nested if Statement (Cont…)

- Output

```
num1 is less than num2
num1 is greater than 45
```

thank
you . . .

Slides are prepared from various sources, such as Book, Internet Links and many more.