Chapter 3: Control Statements

☞Objectives:

- Concept of program control
- Selection statements
- Loop statements
- specific break and continue

Chapter 3: Control Statements

- Selection Statements
  - Using `if` and `if...else`
  - Nested `if` Statements
  - Using `switch` Statements
  - Conditional Operator
- Repetition Statements
  - Looping: `while`, `do`, and `for`
  - Nested loops
  - Using `break` and `continue`

Selection Statements

- `if` Statements (conditional statement)
- `switch` Statements (multi conditional statement)
- Conditional Operators

`if` Statements

```
if (booleanExpression)
{
  statement(s);
}
```

Example:

```
if ((i >= 0) && (i <= 10))
{
  System.out.println("i is an “ +
    “integer between 0 and 10");
}
```

The `if...else` Statement

```
if (booleanExpression)
{
  statement(s)-for-the-true-case;
}
else
{
  statement(s)-for-the-false-case;
}
```

if...else Example

```
if (radius >= 0)
{
  area = radius*radius*PI;
  System.out.println("The area for the “
    + “circle of radius " + radius +
    " is " + area);
}
else
{
  System.out.println("Negative input");
}
```

Nested `if` Statements

Example 3.1: Using Nested `if` Statements

This program reads in number of years and loan amount and computes the monthly payment and total payment. The interest rate is determined by number of years.

```
// TestIfElse.java: Test if-else statements
public class TestIfElse
{
  // Main method
  public static void main(String[] args)
  {
    double annualInterestRate = 0;
    int numOfYears;
    double loanAmount;

    // Enter number of years
    System.out.print(
      "Enter number of years (7, 15 and 30 only): ");
    numOfYears = MyInput.readInt();
```

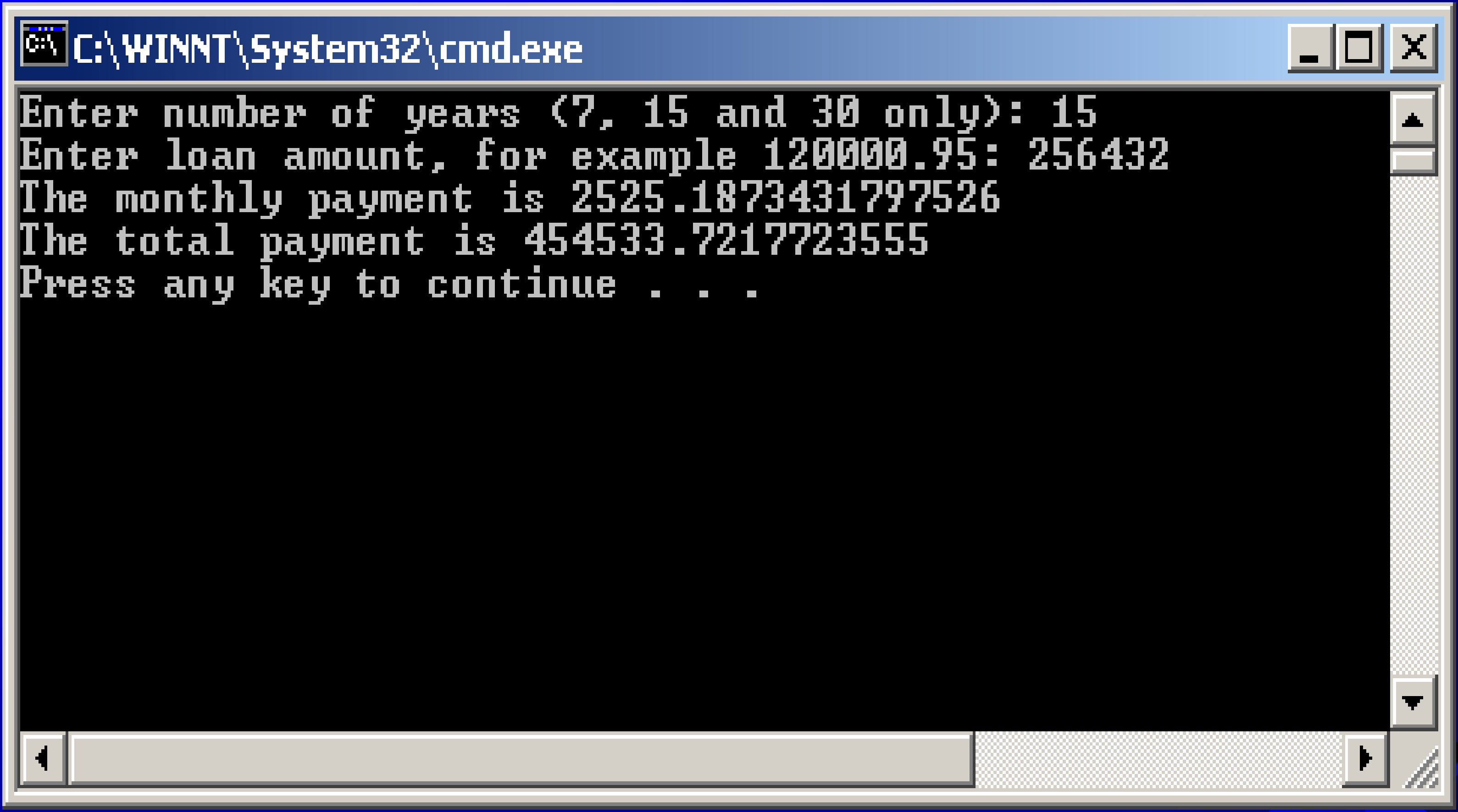
```
// Find interest rate based on year
   if (numOfYears == 7)
     annualInterestRate = 7.25;
   else if (numOfYears == 15)
     annualInterestRate = 8.50;
   else if (numOfYears == 30)
     annualInterestRate = 9.0;
   else
   {
     System.out.println("Wrong number of years");
     System.exit(0);
   }

   // Obtain monthly interest rate
   double monthlyInterestRate = annualInterestRate/1200;
```

```
// Enter loan amount
    System.out.print("Enter loan amount, for example 120000.95: ");
    loanAmount = MyInput.readDouble();

    // Compute mortgage
    double monthlyPayment = loanAmount*monthlyInterestRate/
      (1-(Math.pow(1/(1+monthlyInterestRate), numOfYears*12)));
    double totalPayment = monthlyPayment*numOfYears*12;

    // Display results
    System.out.println("The monthly payment is " + monthlyPayment);
    System.out.println("The total payment is " + totalPayment);
  }
}
```



```
Enter number of years (7, 15 and 30 only): 15
Enter loan amount, for example 120000.95: 256432
The monthly payment is 2525.1873431797526
The total payment is 454533.7217723555
Press any key to continue . . .
```

Conditional Operator

```
if (x > 0)
    y = 1
else
    y = -1;
```

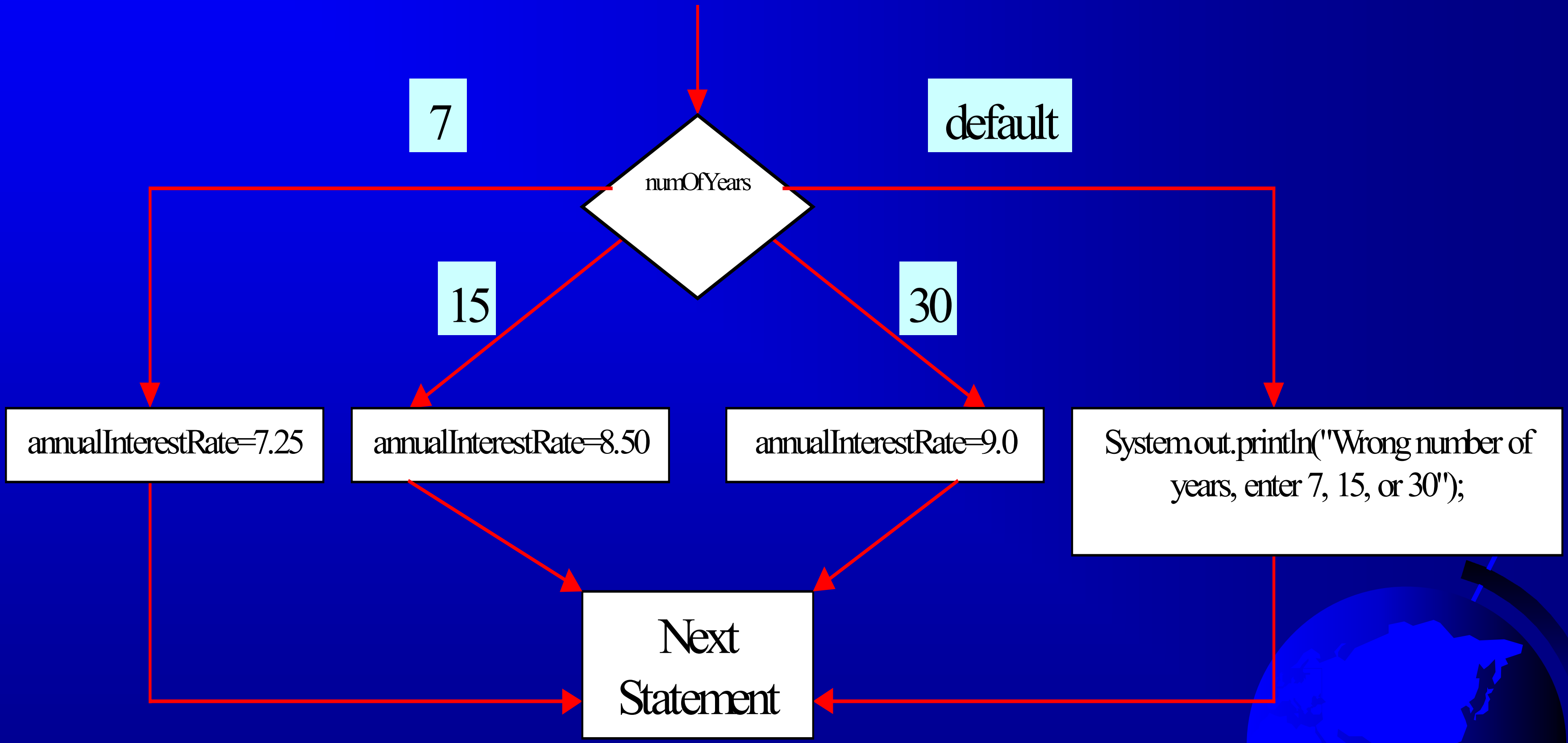
is equivalent to

```
y = (x > 0) ? 1 : -1;
```

switch Statements

```
switch (year)
{
  case 7:  annualInterestRate = 7.25;
           break;
  case 15: annualInterestRate = 8.50;
           break;
  case 30: annualInterestRate = 9.0;
           break;
  default: System.out.println(
   "Wrong number of years, enter 7, 15, or 30");
}
```

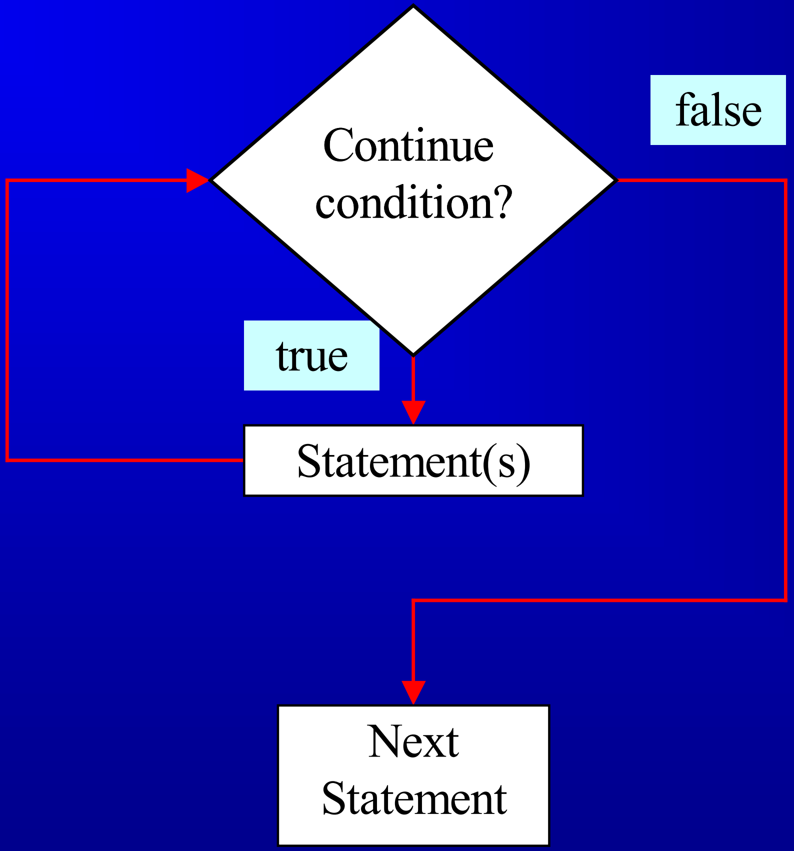
switch Statement Flow Chart



Repetitions

- `while` Loops
- `do` Loops
- `for` Loops
- `break` and `continue`

`while` Loop Flow Chart



while Loops

```
while (continue-condition)
{
  // loop-body;
}
```

Example 3.2: Using while Loops

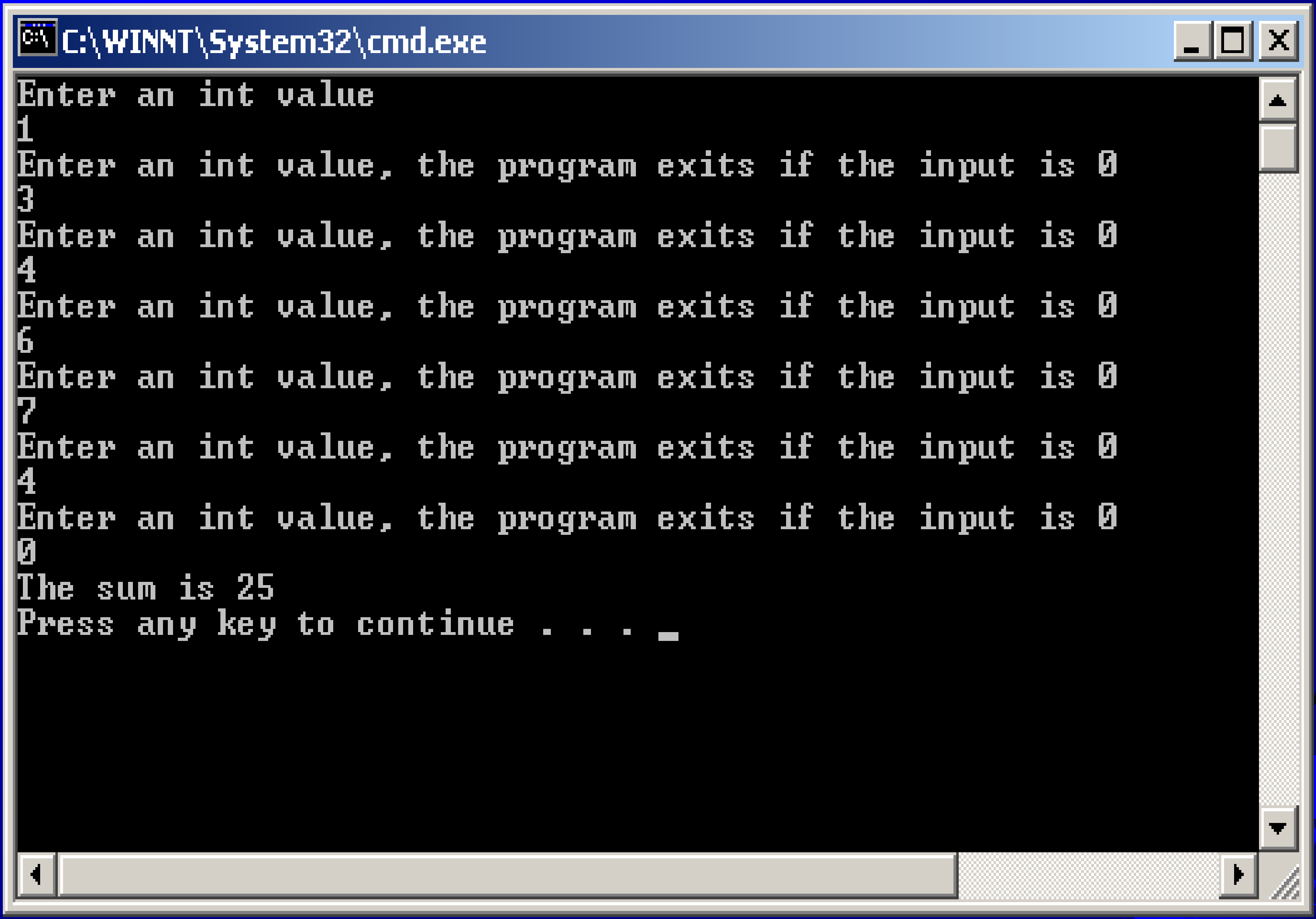
TestWhile.java

```
// TestWhile.java: Test the while loop
public class TestWhile
{
 // Main method
 public static void main(String[] args)
 {
  int data;
  int sum = 0;

  // Read an initial data
  System.out.println("Enter an int value");
  data = MyInput.readInt();
// Keep reading data until the input is 0
   while (data != 0)
   {
     sum += data;

     System.out.println(
       "Enter an int value, the program exits if the input is 0");
     data = MyInput.readInt();
   }

   System.out.println("The sum is " + sum);
  }
}
```

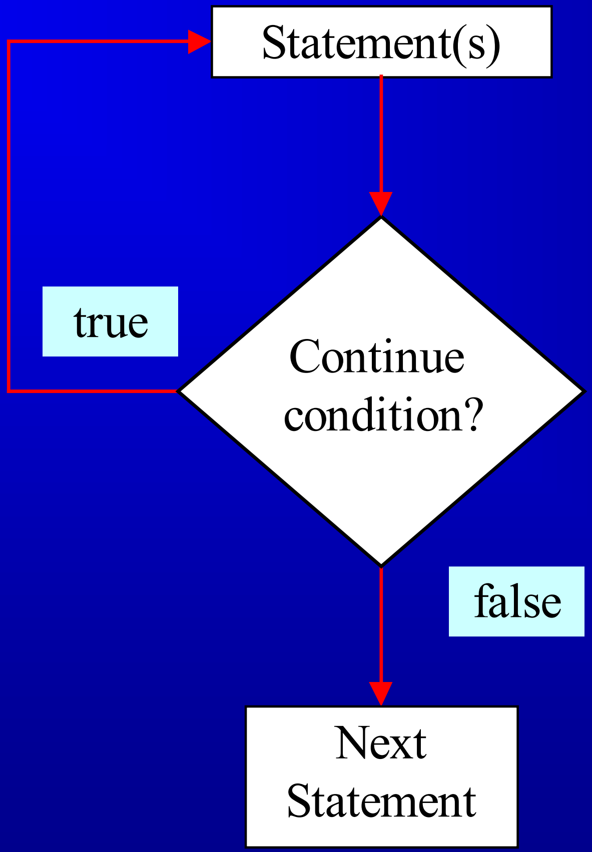


```
Enter an int value
1
Enter an int value, the program exits if the input is 0
3
Enter an int value, the program exits if the input is 0
4
Enter an int value, the program exits if the input is 0
6
Enter an int value, the program exits if the input is 0
7
Enter an int value, the program exits if the input is 0
4
Enter an int value, the program exits if the input is 0
0
The sum is 25
Press any key to continue . . . _
```

do Loops

```
do
{
  // Loop body;
} while (continue-condition)
```

do Loop Flow Chart



for Loops

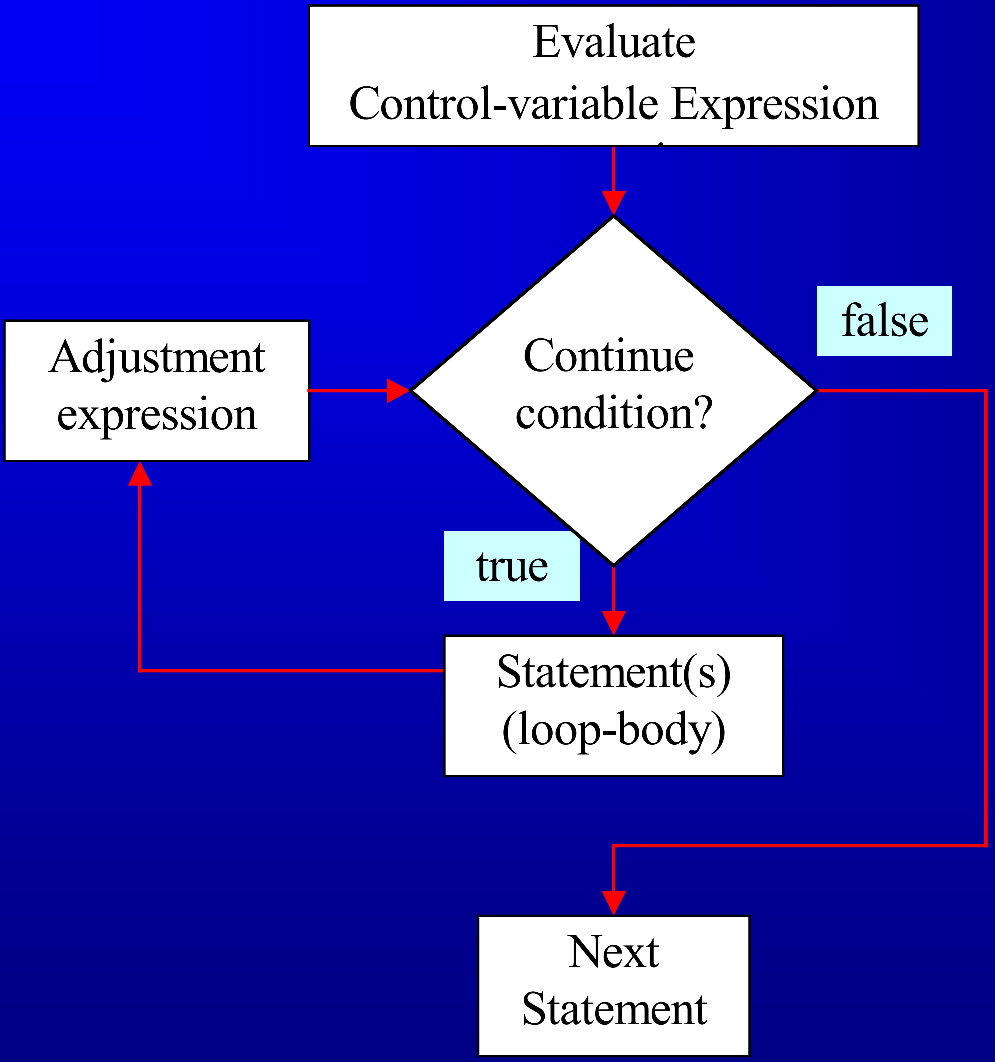
```
for (control-variable-initializer;
      continue-condition; adjustment-statement)
{
   //loop body;
}

int i = 0;
while (i < 100)
{
  System.out.println("Welcome to Java! ” + i);
  i++;
}
```

Example:

```
int i;
for (i = 0; i<100; i++)
{
  System.out.println("Welcome to Java! ” + i);
}
```

`for` Loop Flow Chart



`for` Loop Examples

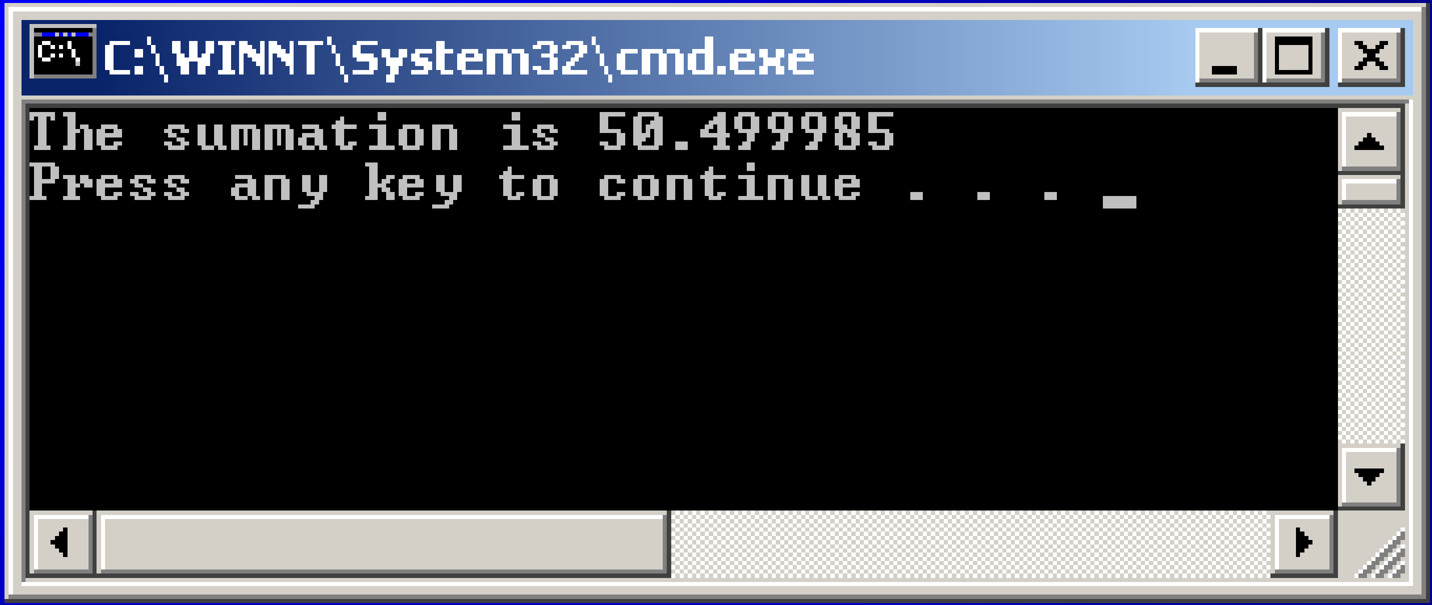
Examples for using the `for` loop:

- Example 3.3: Using for Loops

```
// TestSum.java: Compute sum = 0.01 + 0.02 + … + 1;
public class TestSum
{
  // Main method
  public static void main(String[] args)
  {
    // Initialize sum
    float sum = 0;

    // Keep adding 0.01 to sum
    for (float i=0.01f; i <= 1.0f ; i = i+0.01f)
      sum += i;

    // Display result
    System.out.println("The summation is " + sum);
  }
}
```



```
The summation is 50.499985
Press any key to continue . . . _
```

`for` nested Loop Examples

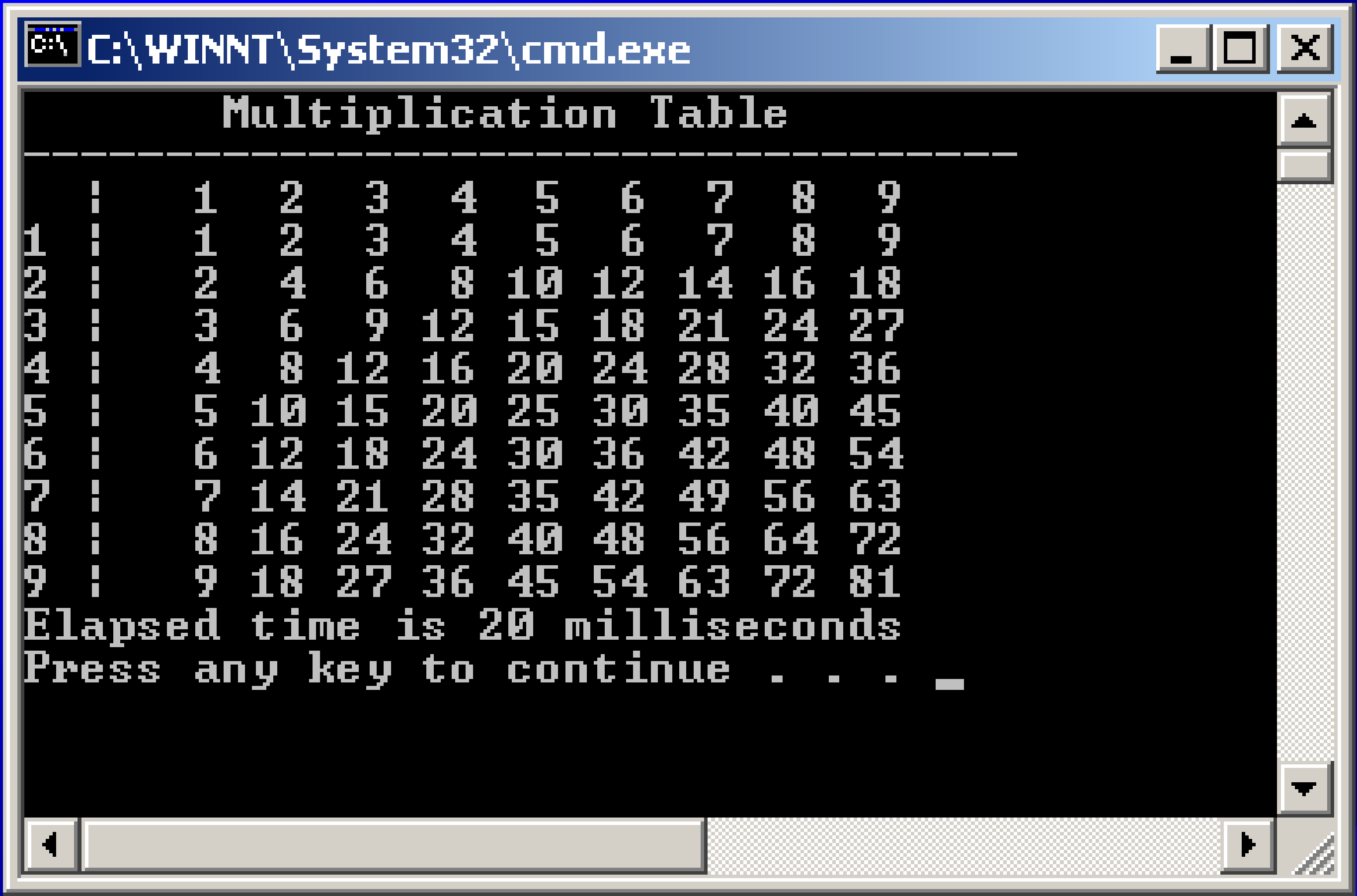
- ☞ Example 3.4: Multiplication Table; Using Nested for Loops

```
// TestMulTable.java: Display a multiplication table
public class TestMulTable
{
  // Main method
  public static void main(String[] args)
  {
    // Get start time
    long startTime = System.currentTimeMillis();
    // Display the table heading
    System.out.println("       Multiplication Table");
    System.out.println("-----------------------------------");

    // Display the number title
    System.out.print("  | ");
    for (int j=1; j<=9; j++)
      System.out.print("  " + j);
    System.out.println(" ");
```

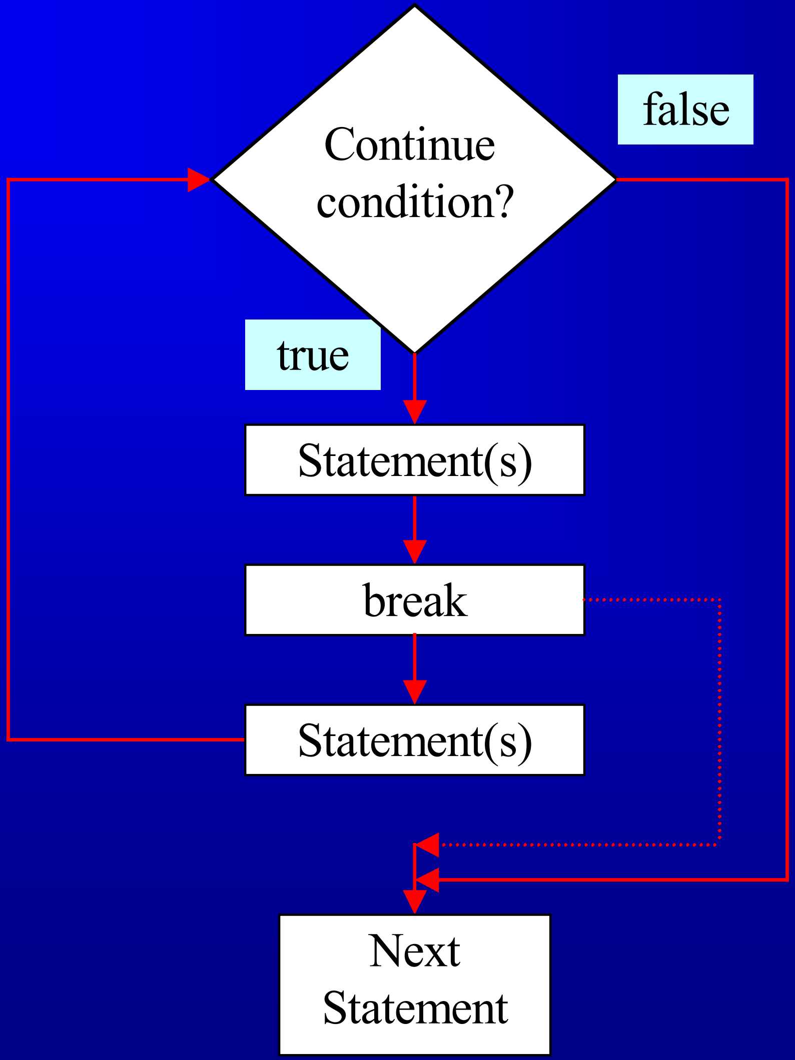
```
// Print table body
    for (int i=1; i<=9; i++)
    {
      System.out.print(i+" | ");
      for (int j=1; j<=9; j++)
      {
        // Display the product and align properly
        if (i*j < 10)
          System.out.print("  " + i*j);
        else
          System.out.print(" " + i*j);
      }
      System.out.println();
    }
```

```
// Get end time
    long endTime = System.currentTimeMillis();
    System.out.println("Elapsed time is " + (endTime - startTime)
      + " milliseconds");
  }
}
```



```
       Multiplication Table
-----------------------------------
  ¦   1  2  3  4  5  6  7  8  9
1 ¦   1  2  3  4  5  6  7  8  9
2 ¦   2  4  6  8 10 12 14 16 18
3 ¦   3  6  9 12 15 18 21 24 27
4 ¦   4  8 12 16 20 24 28 32 36
5 ¦   5 10 15 20 25 30 35 40 45
6 ¦   6 12 18 24 30 36 42 48 54
7 ¦   7 14 21 28 35 42 49 56 63
8 ¦   8 16 24 32 40 48 56 64 72
9 ¦   9 18 27 36 45 54 63 72 81
Elapsed time is 20 milliseconds
Press any key to continue . . .
```

The `break` Keyword

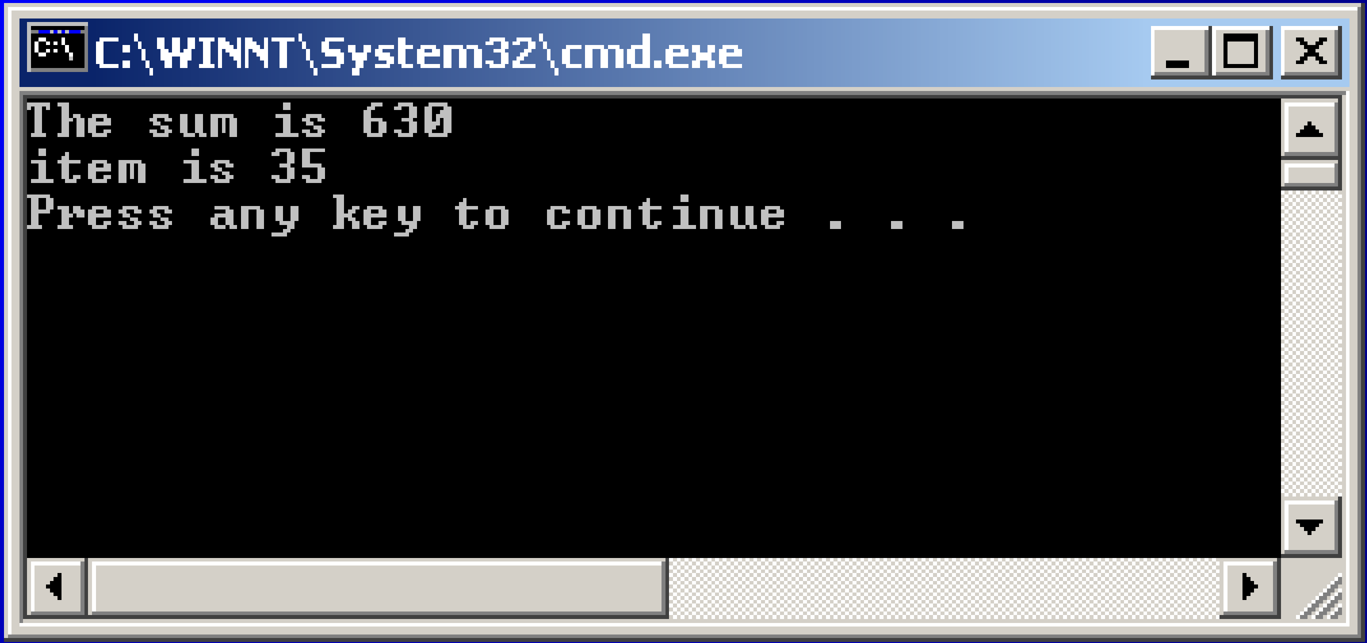


Using `break` and `continue`

Examples for using the `break` keyword:

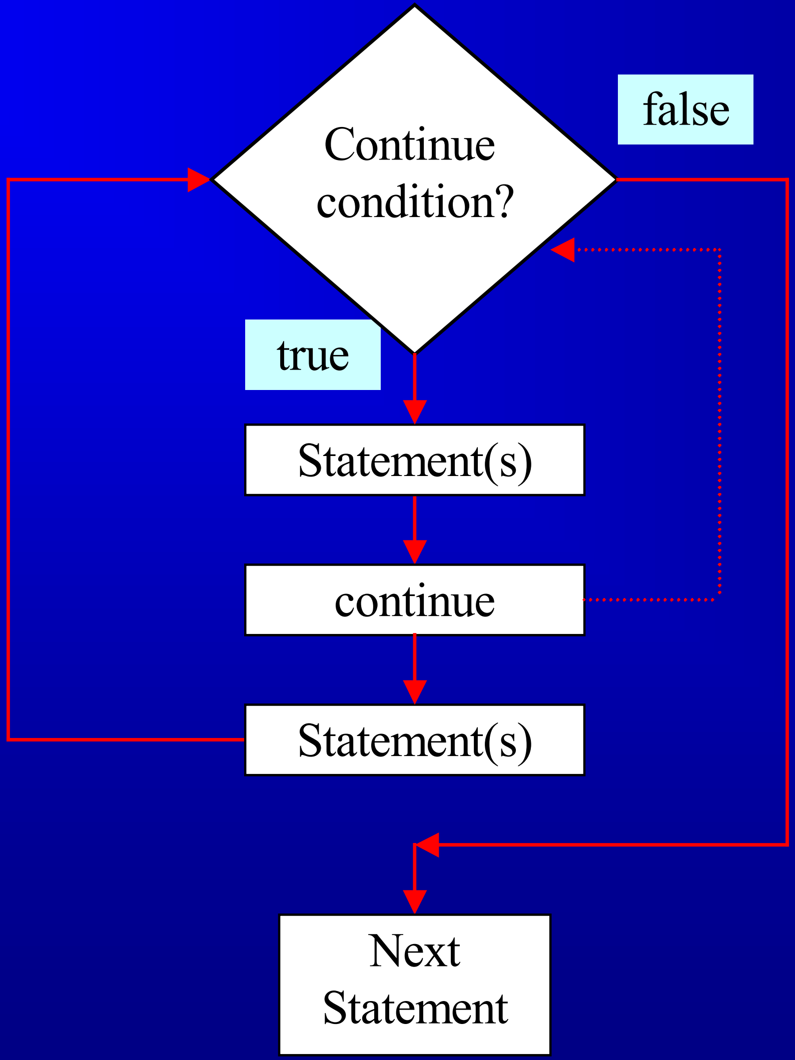
- ☞ Example 3.5: TestBreak.java

```
// TestBreak.java: Test the break keyword in the loop
public class TestBreak
{
  // Main method
  public static void main(String[] args)
  {
    int sum = 0;
    int item = 0;
    while (item < 2000)
    {
      item ++;
      sum += item;
      if (sum >= 600) break;
    }
    System.out.println("The sum is " + sum);
                System.out.println("item is " + item);
  }
}
```



```
The sum is 630
item is 35
Press any key to continue . . .
```

The `continue` Keyword



Continue condition?

false

true

Statement(s)

continue

Statement(s)

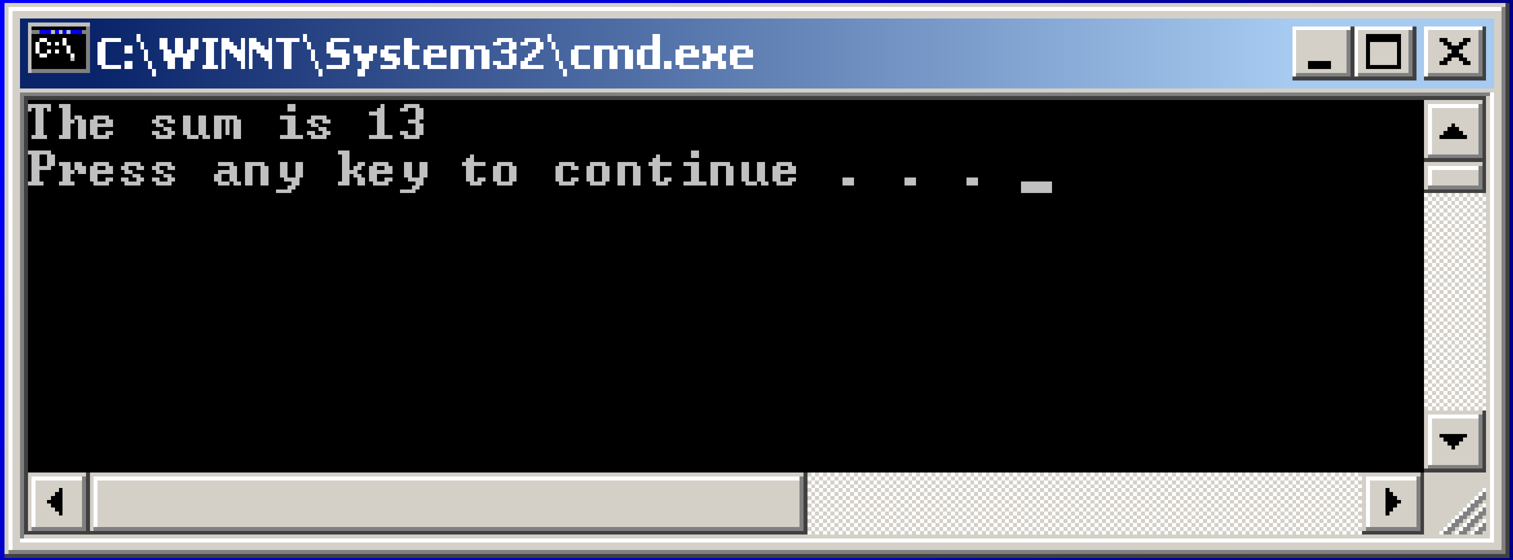
Next Statement

Using `break` and `continue`

Examples for using the `continue` keyword:

☞ Example 3.6: TestContinue.java

```
// TestContinue.java: Test the continue keyword
public class TestContinue
{
  // Main method
  public static void main(String[] args)
  {
    int sum = 0;
    int item = 0;
    while (item < 5)
    {
      item++;
      if (item == 2) continue;
      sum += item;
    }
    System.out.println("The sum is " + sum);
  }
}
```



```
The sum is 13
Press any key to continue . . . _
```

Using Statement Labels and breaking with Labels

```
outer:
  for (int i=1;i<10;i++)
  {
  inner:
     for (int j=1;j<10;j++)
     {
       break outer;
      System.out.println(i*j);
     }
}
```

Case Studies (1)

Example: Commission using while and if statements

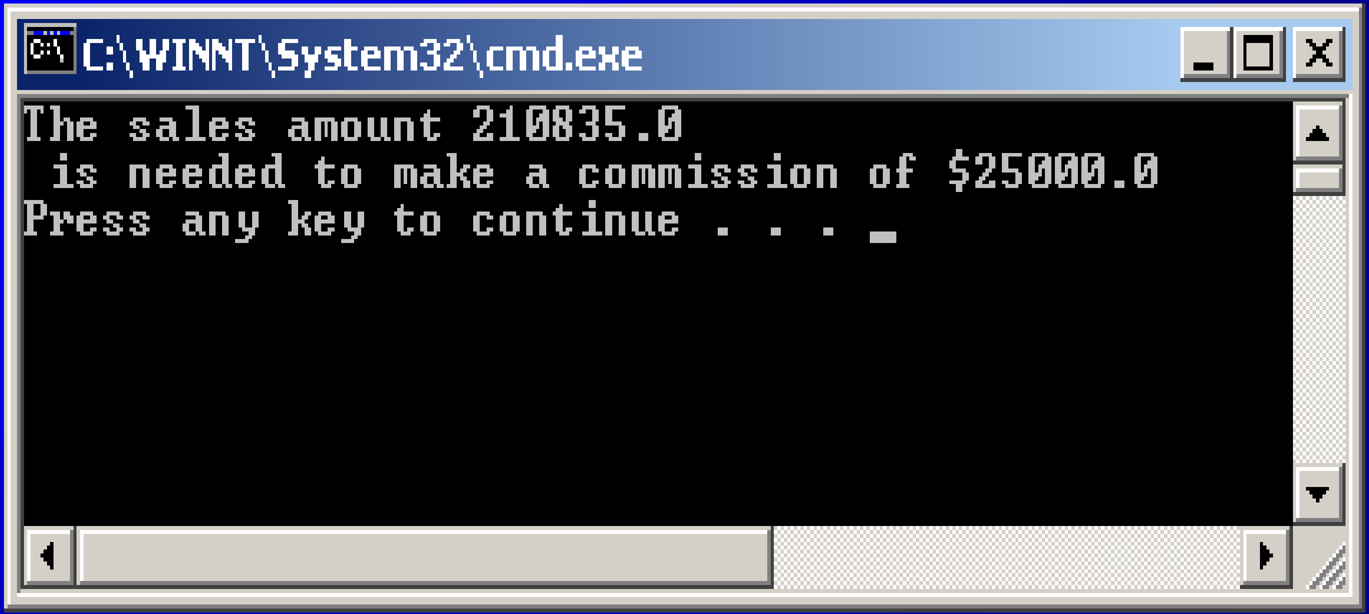
```
// FindSalesAmount.java: Find the sales amount to get the desired
// commission
public class FindSalesAmount
{
  // Main method
  public static void main(String[] args)
  {
    // The commission sought
    final double COMMISSION_SOUGHT = 25000;

    double commission = 0;
    double salesAmount = 1;

    while (commission < COMMISSION_SOUGHT)
    {
      // Compute commission
if (salesAmount >= 10001)
    commission = 5000*0.08 + 5000*0.1
                + (salesAmount-10000)*0.12;
   else if (salesAmount >= 5001)
    commission = 5000*0.08 + (salesAmount-5000)*0.10;
   else
    commission = salesAmount*0.08;

   salesAmount++;
  }

  // Display the sales amount
  System.out.println("The sales amount " + salesAmount +
   " is needed to make a commission of $"
                     + COMMISSION_SOUGHT);
 }
}
```



```
The sales amount 210835.0
 is needed to make a commission of $25000.0
Press any key to continue . . . _
```

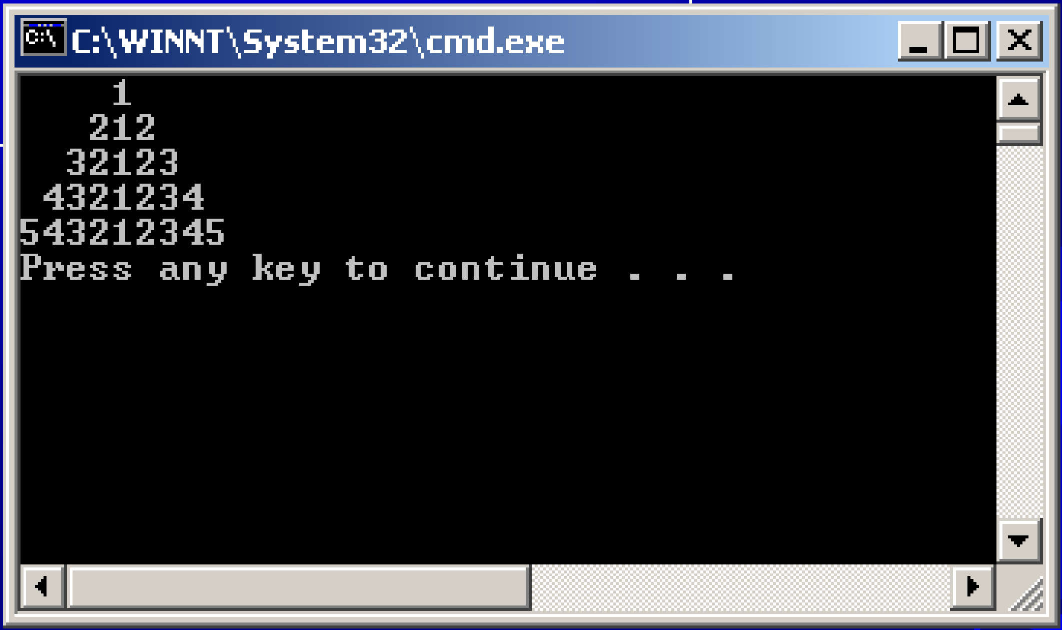
Case Studies (2)

Example: Display a Pyramid using nested for loop

```
public class PrintPyramid
{
  // Main method
  public static void main(String[] args)
  {
    for (int row = 1; row < 6; row++)
    {
      // Print leading spaces
      for (int column = 1; column < 6 - row; column++)
        System.out.print(" ");

      // Print leading numbers
      for (int num = row; num >= 1; num--)
        System.out.print(num);
// Print ending numbers
      for (int num = 2; num <= row; num++)
        System.out.print(num);

      // Start a new line
      System.out.println();
    }
  }
}
```



```
    1
   212
  32123
 4321234
543212345
Press any key to continue . . .
```

Case Studies (3)

Example: Calculate prime number

```
// PrimeNumber.java: Print first 50 prime numbers
public class PrimeNumber
{
  // Main method
  public static void main(String[] args)
  {
    int count = 1; // Count the number of prime numbers
    int number = 2; // A number to be tested for primeness
    boolean isPrime = true;  // If the current number is prime?

    System.out.println("The first 50 prime numbers are \n");
```

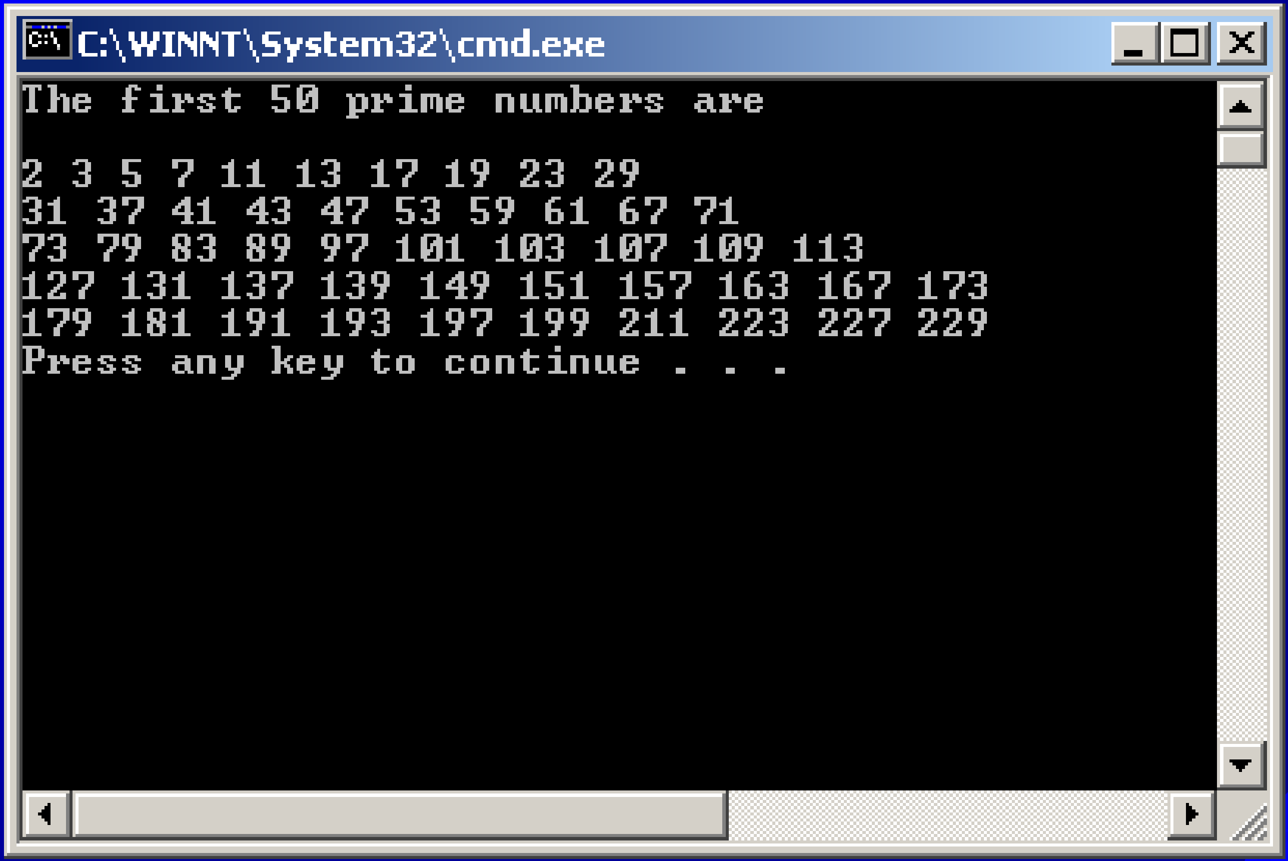
```
// Repeatedly test if a new number is prime
   while (count <= 50)
   {
     // Assume the number is prime
     isPrime = true;

     // Set isPrime to false, if the number is prime
     for (int divisor = 2; divisor <= number/2; divisor++)
     {
       if (number % divisor == 0) // If true, the number is prime
       {
         isPrime = false;
         break;  // Exit the for loop
       }
     }

}
```

```
// Print the prime number and increase the count
      if (isPrime)
      {
        if (count%10 == 0)
        {
          // Print the number and advance to the new line
          System.out.println(number);
        }
        else
          System.out.print(number + " ");
        count++;  // Increase the count
      }

      // Check if the next number is prime
      number++;
    }
  }
```



```
The first 50 prime numbers are

2 3 5 7 11 13 17 19 23 29
31 37 41 43 47 53 59 61 67 71
73 79 83 89 97 101 103 107 109 113
127 131 137 139 149 151 157 163 167 173
179 181 191 193 197 199 211 223 227 229
Press any key to continue . . .
```