Tutorial 9 – **Income Tax Calculator** Application: Introducing the **`switch`** Multiple-Selection Statement

Outline

9.1 Test-Driving the **Income Tax Calculator** Application

9.2 Introducing the **`switch`** Multiple-Selection Statement

7.3 Implicit (promotion) and Explicit (cast) conversion

9.3 Constructing the **Income Tax Calculator** Application

9.4 Wrap-Up

Objectives

- In this tutorial, you will learn to:
  - Use the **`switch`** multiple-selection statement to choose among many alternative actions.
  - Use **`case`** labels to identify the alternative actions in **`switch`** statements.
  - Understand the **`short`** and **`long`** data types to represent small and large integers.

9.1 Test-Driving the **Income Tax Calculator** Application

Application Requirements

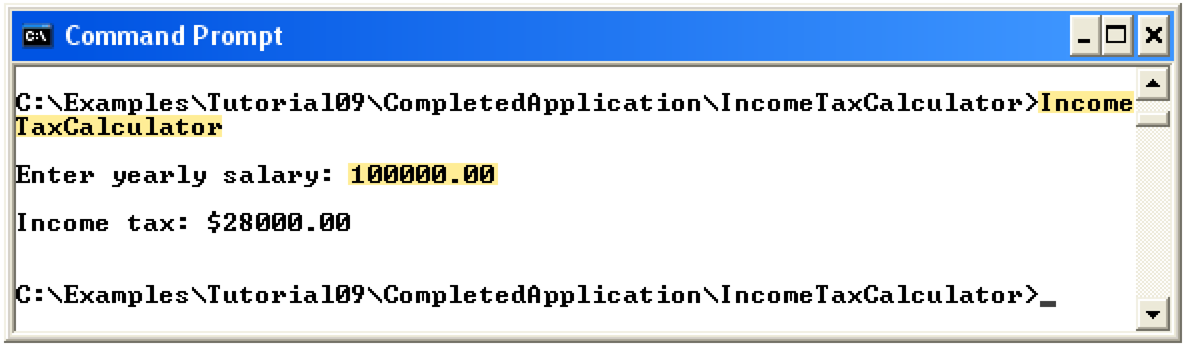
An accounting firm wants an application that estimates the amount of Federal income tax that a client must pay, depending upon that client's annual salary. The application should use the following income ranges and corresponding tax rates:

Under $25,000	=	*15%*	*income*	*tax*
$25,000–74,999	=	*25%*	*income*	*tax*
$75,000–149,999	=	*28%*	*income*	*tax*
$150,000–299,999	=	*33%*	*income*	*tax*
$300,000 and over	=	*35%*	*income*	*tax*

Your application should display an error message if the user enters an annual salary that is less than or equal to zero. [Note: The actual U.S. Federal income tax rates vary based on many factors. For more information, see the information in IRS Form 1040-ES, which is located at ***www.irs.gov/pub/irs-pdf/f1040e03.pdf****. In Exercise 9.12, you will create an application that calculates a closer approximation of an individual's Federal income tax.]*

9.1 Test-Driving the **Income Tax Calculator** Application (Cont.)

Figure 9.3 **Income Tax Calculator** application output for a $100,000 yearly salary.



```
C:\Examples\Tutorial09\CompletedApplication\IncomeTaxCalculator>Income
TaxCalculator

Enter yearly salary: 100000.00

Income tax: $28000.00


C:\Examples\Tutorial09\CompletedApplication\IncomeTaxCalculator>_
```

9.2 Introducing the **switch** Multiple-Selection Statement

The syntax of the `switch` statement is a bit peculiar. Its objective is to check several possible constant values for an expression. Something similar to what we did at the beginning of this section with the concatenation of several if and else if instructions (source: cplusplus.com)

```
switch (expression)
{
  case constant1:
     group of statements 1;
     break;
  case constant2:
     group of statements 2;
     break;
  .
  .
  default:
     default group of statements
}
```

9.2 Introducing the `switch` Multiple-Selection Statement

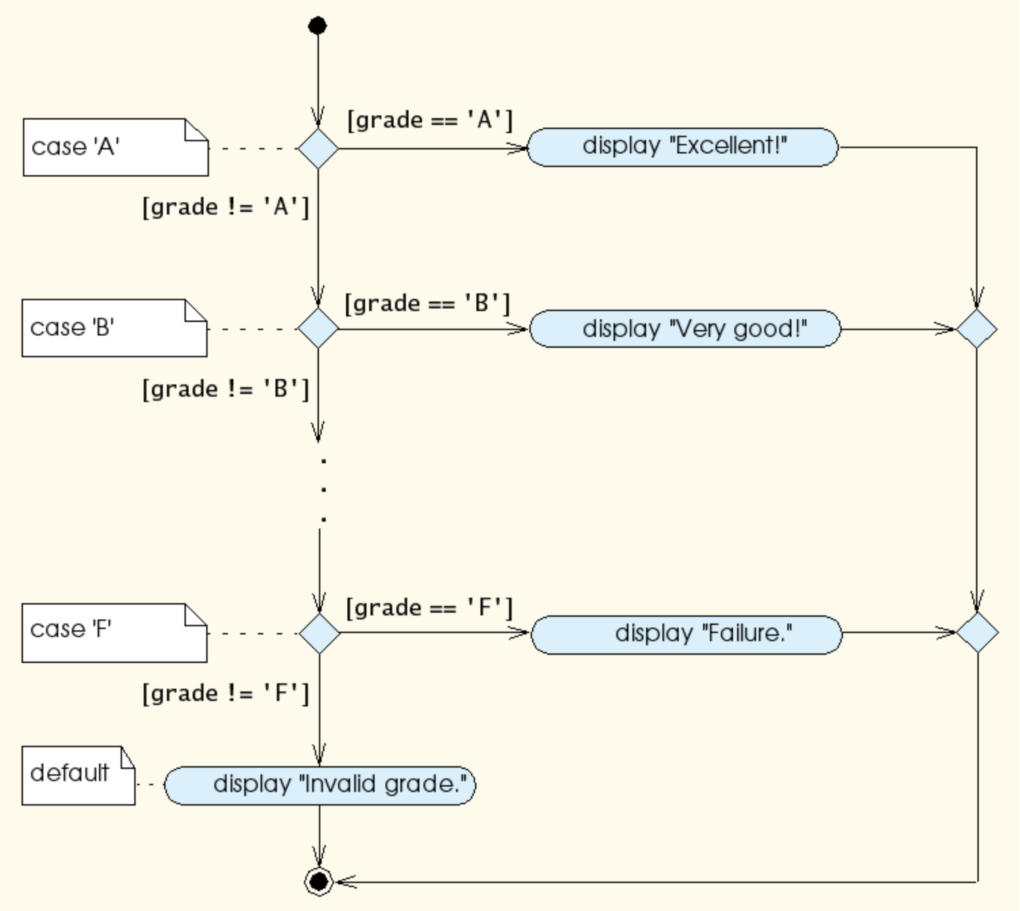
- **`switch`** multiple-selection statements
  - **`switch`** keyword
  - Controlling expression
    - Must be a value of the **`char`**, **`short`**, **`int`** or **`long`** types or a type that can be converted to these types
  - **`case`** labels
    - **`case`** keyword
    - Constant expression
      - Can be a character literal, an integer literal or a variable containing a character or integer constant
      - Cannot be a floating-point value

9.2 Introducing the `switch` Multiple-Selection Statement (Cont.)

- **`break`** statement transfers control to the first statement after the switch statement
  - If a **`case`** label has no **`break`** statement, execution “falls through”
- **`default`** case
  - Executes if no constant expression matches the controlling expression
- A **`switch`** statement cannot contain two **`case`** labels with the same constant expression or two **`default`** cases
- If the controlling expression does not match any constant expression and there is no **`default`** case, none of the statements in the **`switch`** body are executed

9.2 Introducing the **`switch`** Multiple-Selection Statement (Cont.)

Figure 9.4 **`switch`** multiple-selection statement UML activity diagram.

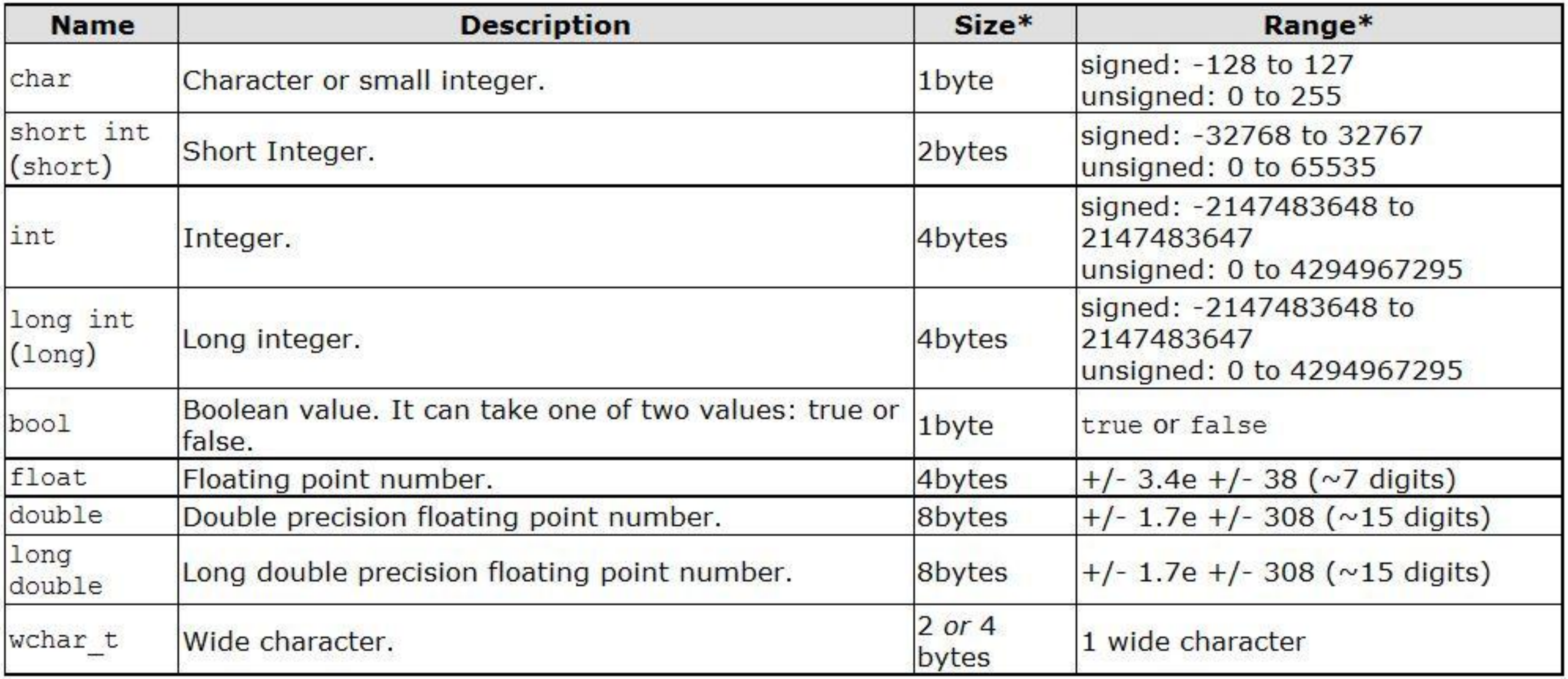


7.3 Implicit (promotion) and Explicit (cast) conversion

- Implicit conversion (promotion)
  - Values are automatically converted to “larger” data types (types at top of the list in Fig. 7.12) when necessary
  - No loss of data
- Explicit conversion (casting)
  - Use unary cast operator **`static_cast<`** *datatype* **`>`**
  - Prevents compiler warnings when converting to a “smaller” data type
  - Converting to a “smaller” data type may cause a loss of data

Fundamental Data Types (*Week 3 Review*)

- Summary of the basic fundamental data types in C++, as well as the range of values that can be represented with each one.



Name	Description	Size*	Range*
`char`	Character or small integer.	1byte	signed: -128 to 127 unsigned: 0 to 255
`short int (short)`	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
`int`	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
`long int (long)`	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
`bool`	Boolean value. It can take one of two values: true or false.	1byte	`true` or `false`
`float`	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
`double`	Double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
`long double`	Long double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
`wchar_t`	Wide character.	2 *or* 4 bytes	1 wide character

7.3 **Implicit (promotion)** and Explicit (cast) conversion

Primitive type	
`long double`	
`double`	
`float`	
`unsigned long int`	(synonymous with `unsigned long`)
`long int`	(synonymous with `long`)
`unsigned int`	(synonymous with `unsigned`)
`int`	
`unsigned short int`	(synonymous with `unsigned short`)
`short int`	(synonymous with `short`)
`unsigned char`	
`char`	
`signed char`	
`bool`	(`false` becomes 0, `true` becomes 1)

Figure 7.12 Promotion hierarchy for C++ primitive types.

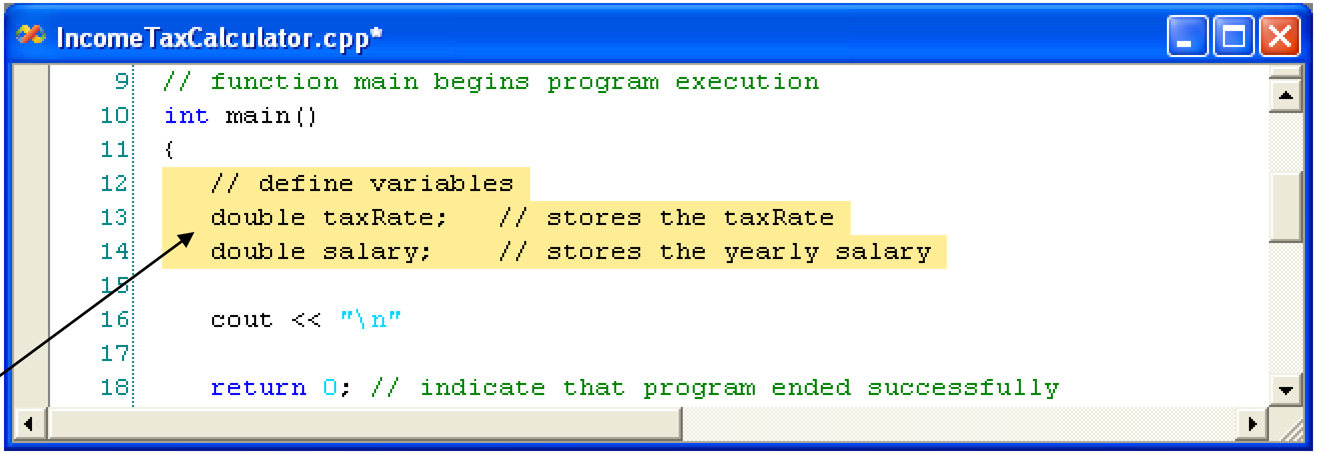
9.3 Constructing the **Income Tax Calculator** Application

```
Prompt the user for and input the yearly salary
If the salary is less than or equal to zero
    Display error message
Else
    Switch based on the integer result of the yearly salary divided by 25,000
        If the result is 0 (salary is less than $25,000)
            Set the tax rate to 15%
        If the result is 1 or 2 (salary is between $25,000 and $74,999, inclusive)
            Set the tax rate to 25%
        If the result is 3, 4 or 5 (salary is between $75,000 and $149,999,
            inclusive)
            Set the tax rate to 28%
        If the result is 6, 7, 8, 9, 10 or 11 (salary is between $150,000 and
            $299,999, inclusive)
            Set the tax rate to 33%
        Otherwise (salary is greater than or equal to $300,000)
            Set the tax rate to 35%
    Calculate and display the income tax
```

Figure 9.5 Pseudocode for the **Income Tax Calculator** application.

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.6 Defining variables to contain the tax rate and the yearly salary.

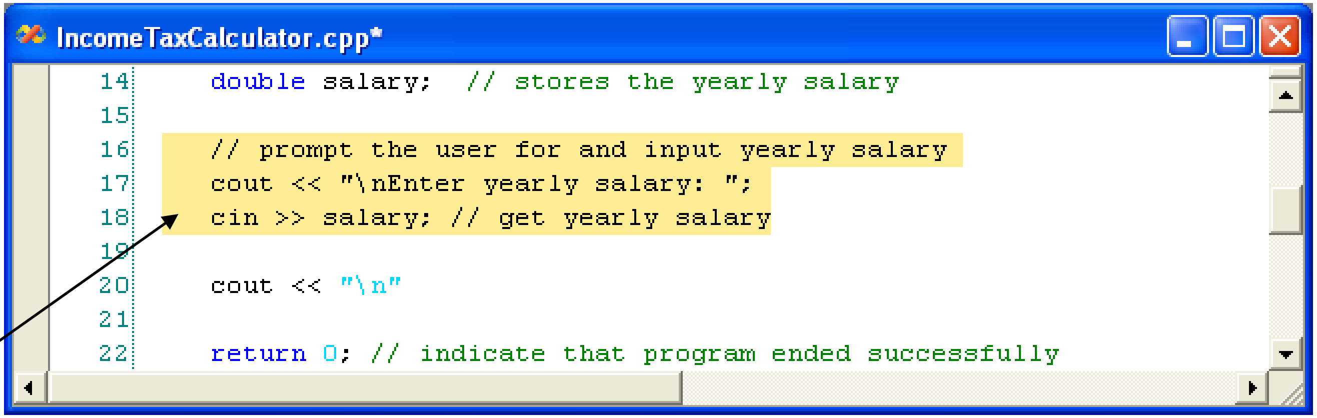


```
// function main begins program execution
int main()
{
   // define variables
   double taxRate;   // stores the taxRate
   double salary;    // stores the yearly salary

   cout << "\n"

   return 0; // indicate that program ended successfully
```

Defining variables to store user input

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.7 Prompting the user for and inputting the yearly salary.

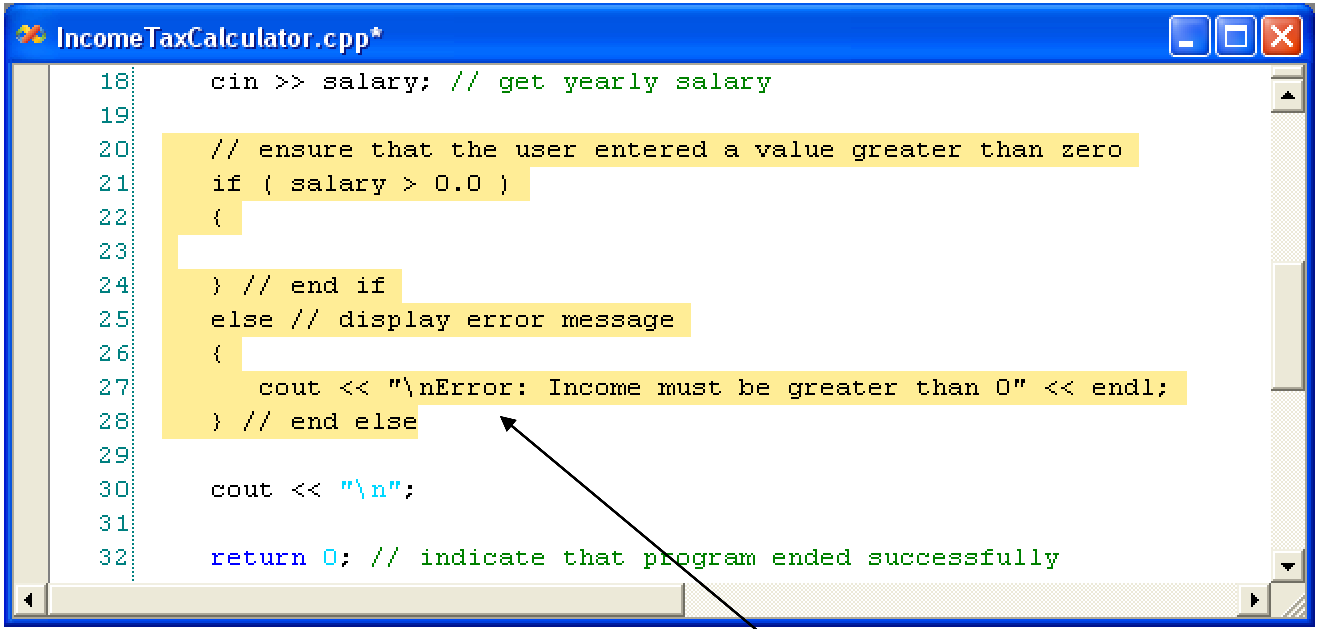


```
   double salary;  // stores the yearly salary

   // prompt the user for and input yearly salary
   cout << "\nEnter yearly salary: ";
   cin >> salary; // get yearly salary

   cout << "\n"

   return 0; // indicate that program ended successfully
```

Prompting the user for and inputting the yearly salary

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.8 Adding an **`if...else`** statement to the function.



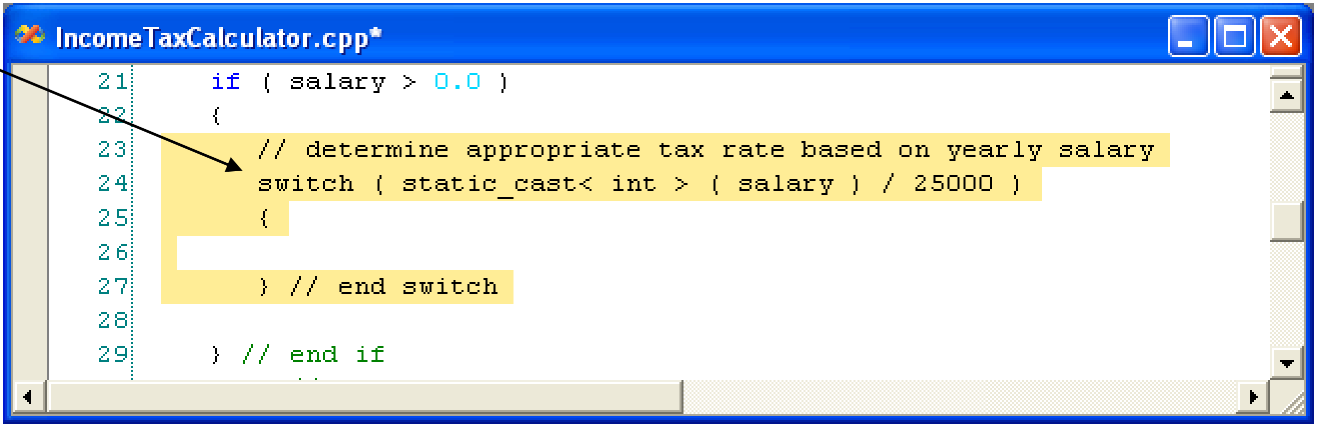
```cpp
   cin >> salary; // get yearly salary

   // ensure that the user entered a value greater than zero
   if ( salary > 0.0 )
   {

   } // end if
   else // display error message
   {
      cout << "\nError: Income must be greater than 0" << endl;
   } // end else

   cout << "\n";

   return 0; // indicate that program ended successfully
```

Displaying an error message if the user enters an invalid salary

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.9 Adding a **`switch`** statement to the function.

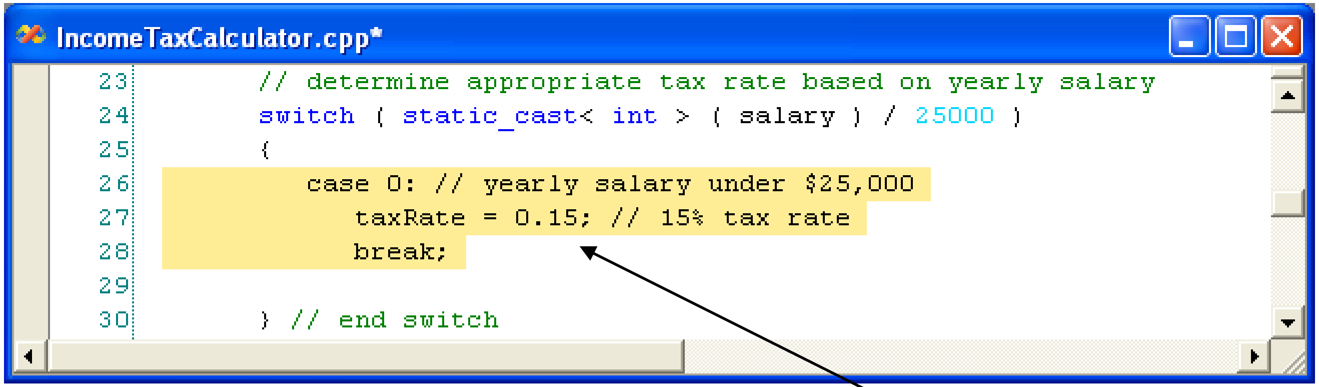
Beginning of the **`switch`** statement



```
if ( salary > 0.0 )
{
   // determine appropriate tax rate based on yearly salary
   switch ( static_cast< int > ( salary ) / 25000 )
   {

   } // end switch

} // end if
```

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.10 Adding a **`case`** label to the switch statement.

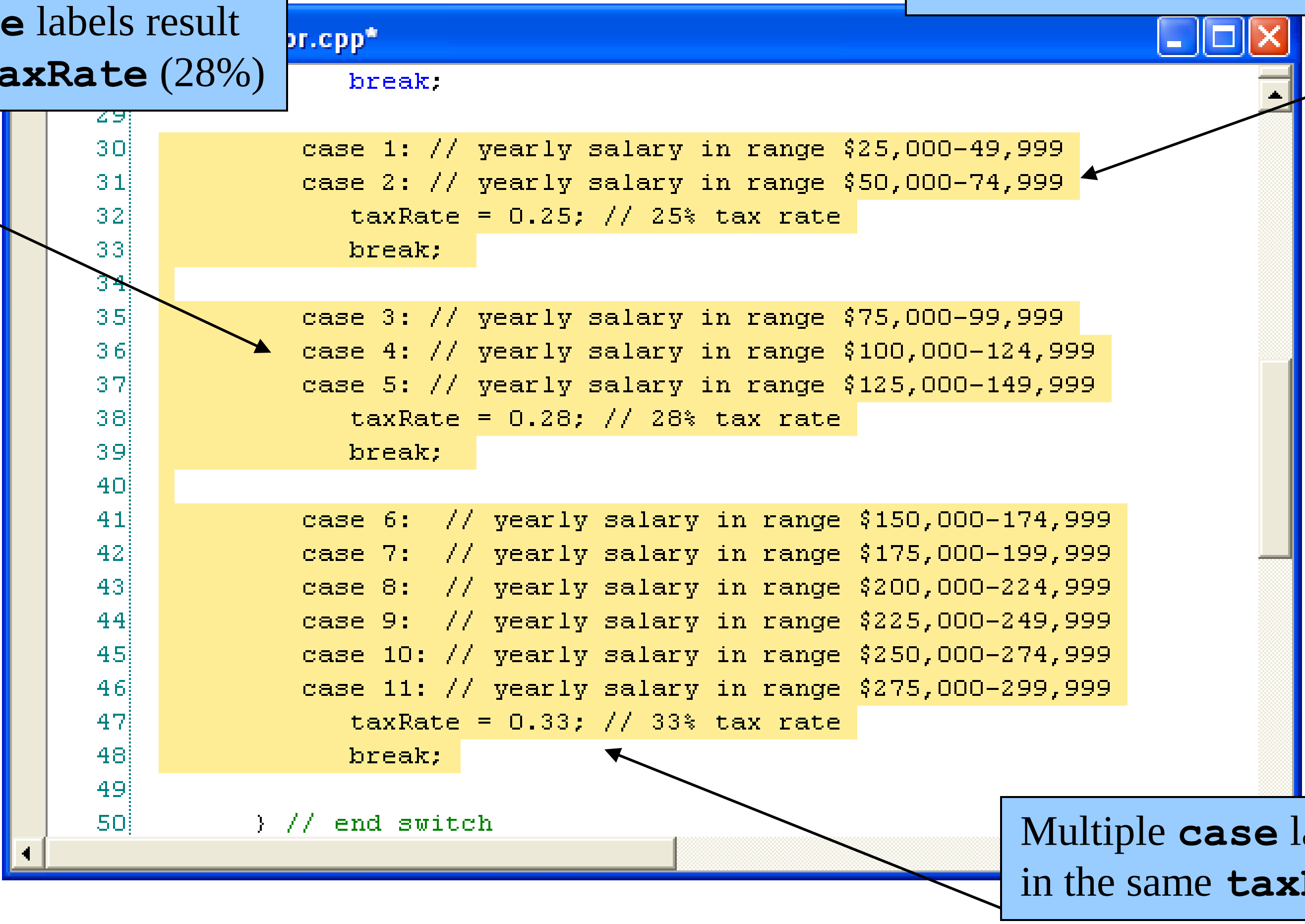


```
      // determine appropriate tax rate based on yearly salary
      switch ( static_cast< int > ( salary ) / 25000 )
      {
         case 0: // yearly salary under $25,000
            taxRate = 0.15; // 15% tax rate
            break;

      } // end switch
```

Sets the tax rate to 15% if **`salary`** is less than **`25000`**

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.11 Finishing the **`switch`** statement.



```
            break;

         case 1: // yearly salary in range $25,000-49,999
         case 2: // yearly salary in range $50,000-74,999
            taxRate = 0.25; // 25% tax rate
            break;

         case 3: // yearly salary in range $75,000-99,999
         case 4: // yearly salary in range $100,000-124,999
         case 5: // yearly salary in range $125,000-149,999
            taxRate = 0.28; // 28% tax rate
            break;

         case 6:  // yearly salary in range $150,000-174,999
         case 7:  // yearly salary in range $175,000-199,999
         case 8:  // yearly salary in range $200,000-224,999
         case 9:  // yearly salary in range $225,000-249,999
         case 10: // yearly salary in range $250,000-274,999
         case 11: // yearly salary in range $275,000-299,999
            taxRate = 0.33; // 33% tax rate
            break;

      } // end switch
```

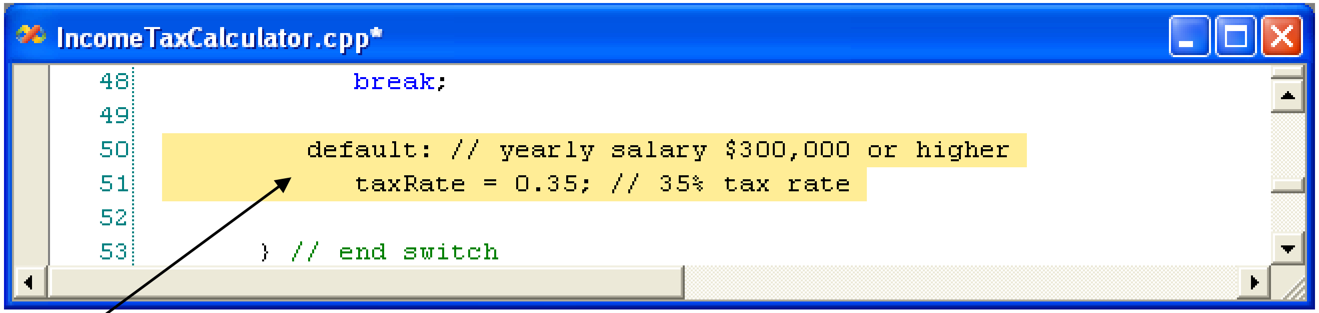
Sets tax rate to 25% if **`salary`** is between **`25000`** and **`74999`**

Multiple **`case`** labels result in the same **`taxRate`** (28%)

Multiple **`case`** labels result in the same **`taxRate`** (33%)

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.12 Adding a **`default`** case to the **`switch`** statement.



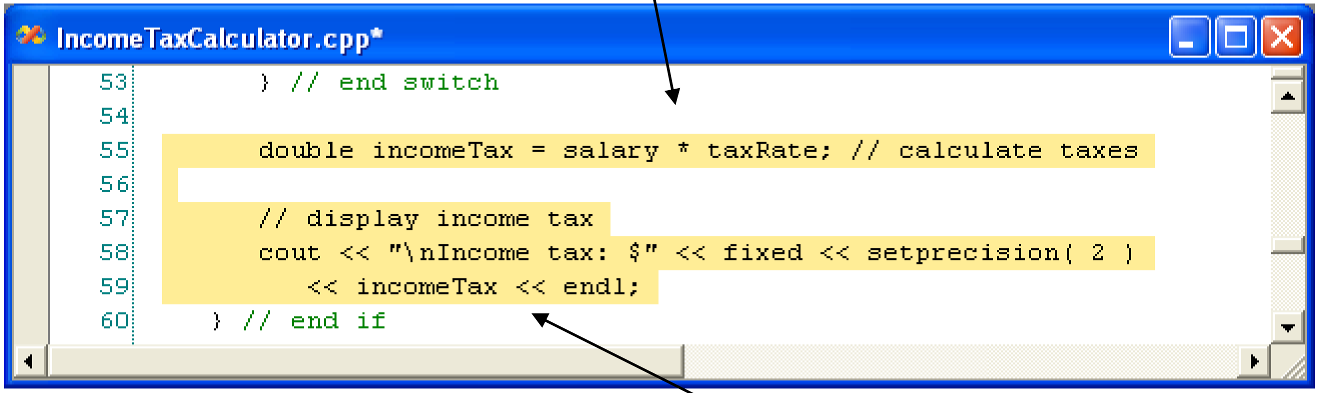
```
      break;

   default: // yearly salary $300,000 or higher
      taxRate = 0.35; // 35% tax rate

} // end switch
```

`default` case at the end of the **`switch`** statement

- When the **`default`** case is at the end of the **`switch`** body, a **`break`** statement is unnecessary

9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.13 Calculating and displaying the income tax.



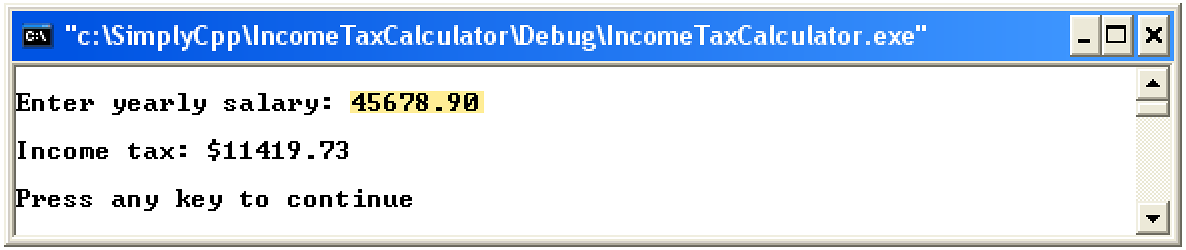
```
      } // end switch

      double incomeTax = salary * taxRate; // calculate taxes

      // display income tax
      cout << "\nIncome tax: $" << fixed << setprecision( 2 )
         << incomeTax << endl;
   } // end if
```

Calculating the income tax

Displaying the income tax

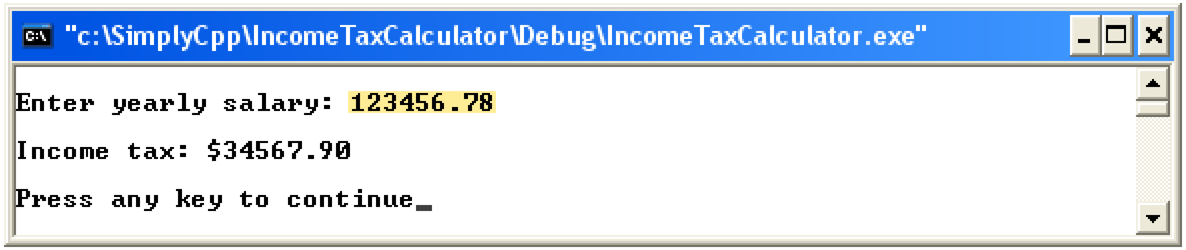
9.3 Constructing the **Income Tax Calculator** Application (Cont.)

Figure 9.14 Completed **Income Tax Calculator** application.



"c:\SimplyCpp\IncomeTaxCalculator\Debug\IncomeTaxCalculator.exe"

```
Enter yearly salary: 45678.90
Income tax: $11419.73
Press any key to continue
```



"c:\SimplyCpp\IncomeTaxCalculator\Debug\IncomeTaxCalculator.exe"

```
Enter yearly salary: 123456.78
Income tax: $34567.90
Press any key to continue_
```

IncomeTaxCalculator.cpp (1 of 4)

```cpp
// Tutorial 9: IncomeTaxCalculator.cpp
// Calculates a person's Federal income tax
// depending on that person's salary.
#include <iostream> // required to perform C++ stream I/O
#include <iomanip>  // required for parameterized stream manipulators

using namespace std; // for accessing C++ Standard Library members

// function main begins program execution
int main()
{
   // define variables
   double taxRate; // tax rate
   double salary;  // yearly salary

   // prompt user for and input yearly salary
   cout << "\nEnter yearly salary: ";
   cin >> salary; // get yearly salary

   // ensure that the user entered a value greater than zero
   if ( salary > 0 )
   {
```

Define variables to store user input

Prompt the user for and input the yearly salary

```cpp
   // determine appropriate tax rate based on yearly salary
   switch ( static_cast< int > ( salary ) / 25000 )
   {
      case 0: // yearly salary under $25,000
         taxRate = 0.15; // 15% tax rate
         break;

      case 1: // yearly salary in range $25,000-49,999
      case 2: // yearly salary in range $50,000-74,999
         taxRate = 0.25; // 25% tax rate
         break;

      case 3: // yearly salary in range $75,000-99,999
      case 4: // yearly salary in range $100,000-124,999
      case 5: // yearly salary in range $125,000-149,999
         taxRate = 0.28; // 28% tax rate
         break;

```

Beginning of the **switch** statement

IncomeTaxCalculator.cpp (2 of 4)

Set the tax rate to 15% if **salary** is less than **25000**

Set the tax rate to 25% if **salary** is between **25000** and **74999**

Set the tax rate to 28% if **salary** is between **75000** and **149999**

```cpp
         case 6:  // yearly salary in range $150,000-174,999
         case 7:  // yearly salary in range $175,000-199,999
         case 8:  // yearly salary in range $200,000-224,999
         case 9:  // yearly salary in range $225,000-249,999
         case 10: // yearly salary in range $250,000-274,999
         case 11: // yearly salary in range $275,000-299,999
            taxRate = 0.33; // 33% tax rate
            break;

         default: // yearly salary $300,000 or higher
            taxRate = 0.35; // 35% tax rate

      } // end switch

      double incomeTax = salary * taxRate; // calculate taxes

      // display income tax
      cout << "\nIncome tax: $" << fixed << setprecision( 2 )
           << incomeTax << endl;
   } // end if
```

IncomeTaxCalculator.cpp (3 of 4)

Set the tax rate to 33% if `salary` is between `150000` and `299999`

`default` case at the end of the `switch` statement

Calculate the income tax

Display the income tax

```
   else // display error message
   {
      cout << "\nError: Income must be greater than 0" << endl;
   } // end else

   cout << "\n";

   return 0; // indicate that program ended successfully

} // end function main
```

Display an error message if annual salary is not greater than zero

IncomeTaxCalculator.cpp (4 of 4)

Lab and Homework Assignment

- Tutorial 4 − Wage Calculator. *Turn in annotated source file with your own comments.*
- Answer and Turn-in Tutorial 4 Questions 4.1, 4.2, 4.7, 4.8, 4.9, 4.10 at the end of the Chapter 4 plus Tutorial 9 Questions 9.1, 9.2, 9.5, 9.7, 9.10 at the end of Chapter 9. *Always write the question followed by the answer. Remember to highlight the answer.*
- Exercises 4.11, 9.11, and 4.17
- Due next Wednesday