CIS 22A – Lecture 8

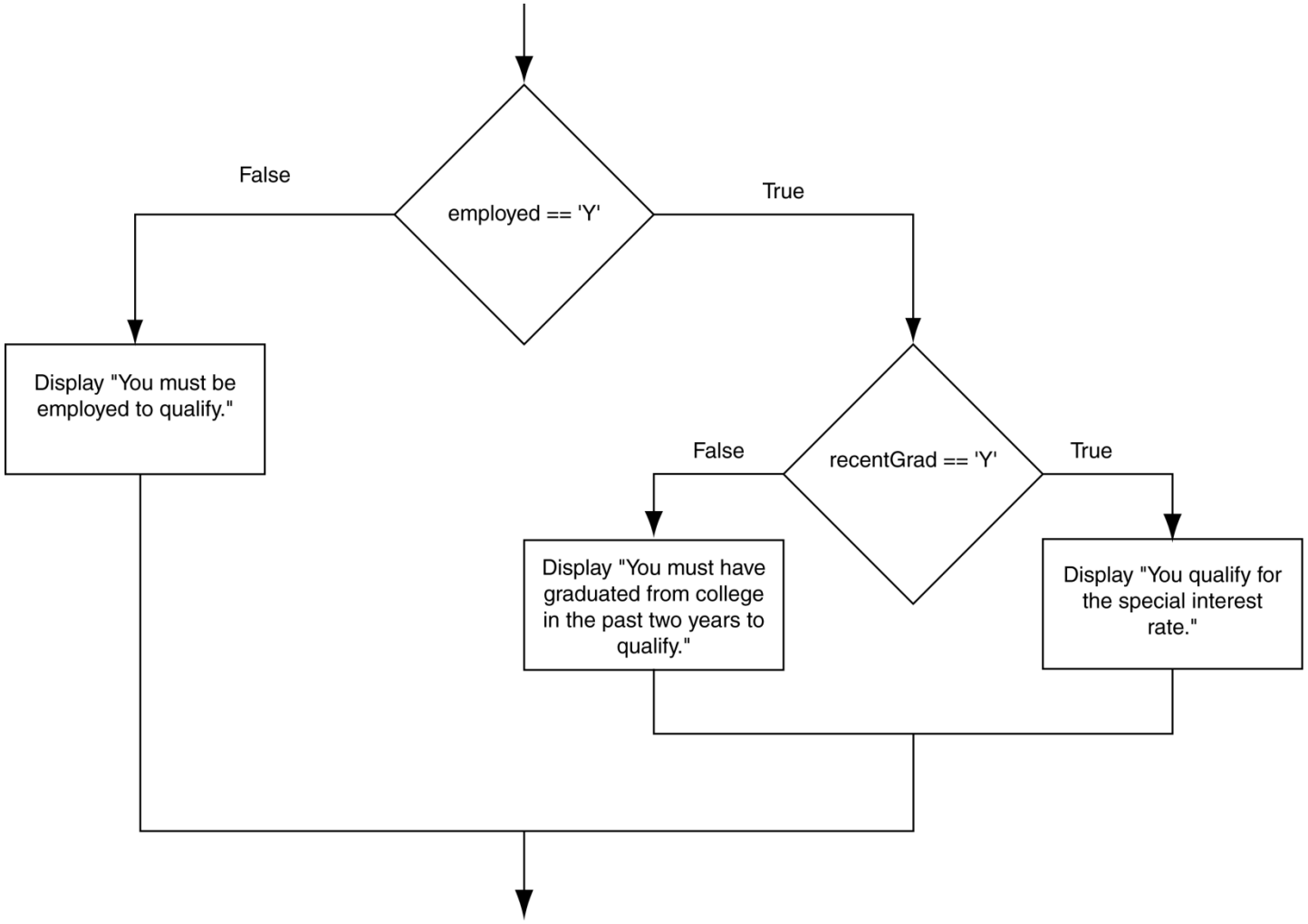
Manish Goel

Nested `if` Statements

- An `if` statement inside another `if` statement
- Nested `if` statements can be used to test more than one condition
- Remember – use indentation and `{ }` to ensure proper nesting

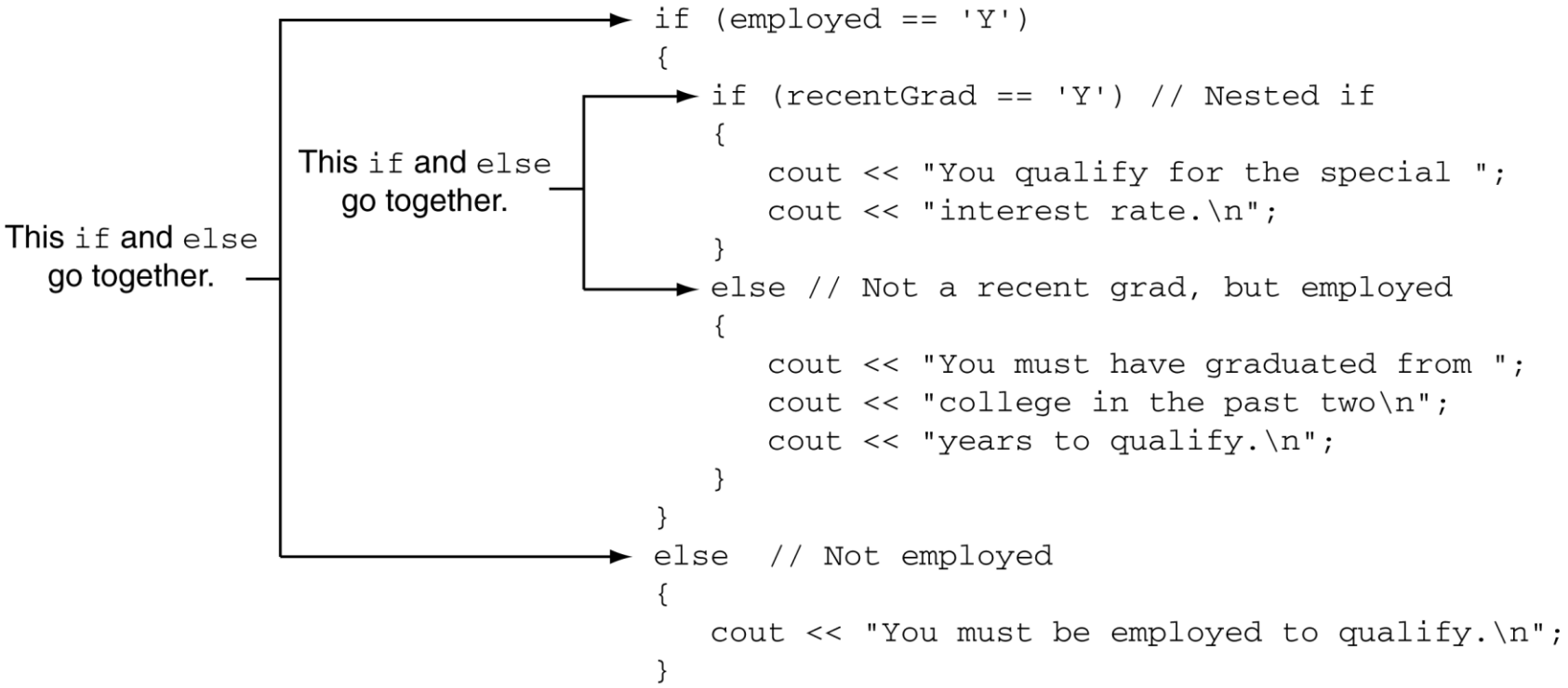
```
// Determine the user's loan qualifications.
if (employed == 'Y')
{
   if (recentGrad == 'Y') //Nested if
   {
      cout << "You qualify for the special ";
      cout << "interest rate.\n";
   }
}
```

Flowchart for a Nested `if` Statement



Nested `if/else` Statements

- Another example, from Program 4-1



Logical Operators

- Used to create complex relational expressions by combining other relational expressions

`&&`	AND	New relational expression is true if both expressions are true
`||`	OR	New relational expression is true if either expression is true
`!`	NOT	Reverses the value of an expression – true expression becomes false, and false becomes true

```
Example:   int x = 12, y = 5, z = -4;
```

`(x > y) && (y > z)`	`true`
`(x > y) && (z > y)`	`false`
`(x <= z) || (y == z)`	`false`
`(x <= z) || (y != z)`	`true`
`!(x >= z)`	`false`

The logical **&&** operator in Program 4-15

```
21    // Determine the user's loan qualifications.
22    if (employed == 'Y' && recentGrad == 'Y')
23    {
24       cout << "You qualify for the special "
25            << "interest rate.\n";
26    }
27    else
28    {
29       cout << "You must be employed and have\n"
30            << "graduated from college in the\n"
31            << "past two years to qualify.\n";
32    }
```

The logical || Operator in Program 4-16

```
23    // Determine the user's loan qualifications.
24    if (income >= MIN_INCOME || years > MIN_YEARS)
25       cout << "You qualify.\n";
26    else
27    {
28       cout << "You must earn at least $"
29            << MIN_INCOME << " or have been "
30            << "employed more than " << MIN_YEARS
31            << " years.\n";
32    }
```

The logical ! Operator in Program 4-17

```
23    // Determine the user's loan qualifications.
24    if (!(income >= MIN_INCOME || years > MIN_YEARS))
25    {
26       cout << "You must earn at least $"
27            << MIN_INCOME << " or have been "
28            << "employed more than " << MIN_YEARS
29            << " years.\n";
30    }
31    else
32       cout << "You qualify.\n";
```

Logical Operator-Notes

- `!` has highest precedence, followed by `&&`, then `||`
- If the value of an expression can be determined by evaluating just the sub-expression on left side of a logical operator, then the sub-expression on the right side will not be evaluated (*short circuit evaluation*)
- Checking Numeric Ranges with Logical Operators
  - Used to test to see if a value falls **inside** a range:

```
if (grade >= 0 && grade <= 100)
   cout << "Valid grade";
```

  - Can also test to see if value falls **outside** of range:

```
if (grade <= 0 || grade >= 100)
   cout << "Invalid grade";
```

  - Cannot use mathematical notation:

```
if (0 <= grade <= 100) //doesn't work!
```

Comparing Characters

- Characters are compared using their ASCII values
- 'A' < 'B'
  - The ASCII value of 'A' (65) is less than the ASCII value of 'B'(66)
- '1' < '2'
  - The ASCII value of '1' (49) is less than the ASCI value of '2' (50)
- Lowercase letters have higher ASCII codes than uppercase letters, so 'a' > 'Z'

Relational Operators Compare Characters in Program 4-20

```
10    // Get a character from the user.
11    cout << "Enter a digit or a letter: ";
12    ch = cin.get();
13
14    // Determine what the user entered.
15    if (ch >= '0' && ch <= '9')
16       cout << "You entered a digit.\n";
17    else if (ch >= 'A' && ch <= 'Z')
18       cout << "You entered an uppercase letter.\n";
19    else if (ch >= 'a' && ch <= 'z')
20       cout << "You entered a lowercase letter.\n";
21    else
22       cout << "That is not a digit or a letter.\n";
```

Comparing `string` Objects

- Like characters, strings are compared using their ASCII values

```
string name1 = "Mary";
string name2 = "Mark";

name1 > name2   // true
name1 <= name2 // false
name1 != name2  // true

name1 < "Mary Jane" // true
```

The characters in each string must match before they are equal

Relational Operators Compare Strings in Program 4-21

```
26      // Determine and display the correct price
27      if (partNum == "S-29A")
28         cout << "The price is $" << PRICE_A << endl;
29      else if (partNum == "S-29B")
30         cout << "The price is $" << PRICE_B << endl;
31      else
32         cout << partNum << " is not a valid part number.\n";
```