switch statement

- syntax
- example
- vs. else-if
• Captures a particular common form of nested-if pattern

• Name “switch” suggests selecting an option based on a central setting
switch syntax

switch ( <case-selector-expr> )
{
    case <label1>: <statement(s)1>
    case <label2>: <statement(s)2>
    ...
    case <labelN>: <statement(s)N>
    default: <statements(s)M>
}

integral

integral
unique
constant

optional
int vehicleClass; double toll; string type;
cout << "Enter vehicle class: ";
cin >> vehicleClass;
if ( vehicleClass == 1 ) {
    type = "Passenger car";
    toll = 0.50;
} else if ( vehicleClass == 2 ) {
    type = "Bus";
    toll = 1.50;
} else if ( vehicleClass == 3 ) {
    type = "Truck";
    toll = 2.00;
} else {
    type = "Unknown vehicle class!";
}

if ( vehicleClass == 1 ) {
    type = "Passenger car";
    toll = 0.50;
} else if ( vehicleClass == 2 ) {
    type = "Bus";
    toll = 1.50;
} else if ( vehicleClass == 3 ) {
    type = "Truck";
    toll = 2.00;
} else {
    type = "Unknown vehicle class!";
}

switch ( vehicleClass ) {
    case 1: type = "Passenger car";
            toll = 0.50;
            break;
    case 2: type = "Bus";
            toll = 1.50;
            break;
    case 3: type = "Truck";
            toll = 2.00;
            break;
    default: type = "Unknown vehicle class!";
}

break

• Used to exit a switch statement
• (Also exits other control structures, such as loops, …)

```c
switch ( vehicleClass )
{
    case 1:
        type = "Passenger car";
        toll = 0.50;
        break;
    case 2:
        type = "Bus";
        toll = 1.50;
        break;
    ...
}
```

With no “break”, the switch statement continues to the code in the next case!
vehicleClass = 1;
switch (vehicleClass)
{
    case 1: type = "Passenger car";  
            toll = 0.50;
    case 2: type = "Bus";            
            toll = 1.50;
    case 3: type = "Truck";         
            toll = 2.00; 
            break; 
    default: type = "Unknown vehicle class!"; 
}

cout << type << endl;

What prints?

Truck
vehicleClass = 1;
switch ( vehicleClass )
{
    case 1 : cout << “Passenger car” << endl;
            toll = 0.50;
    case 2 : cout << “Bus” << endl;
            toll = 1.50;
    case 3 : cout << “Truck” << endl;
            toll = 2.00;
            break;
    default : cout << “Unknown vehicle class!” << endl;
}

What prints?

Passenger car
Bus
Truck
Switch vs. if-then-else

- **Readability**
  - switch table format very clear

- **Generality**
  - switch syntax limits applicability
    - (many things that *if* can do that *switch* can’t)
    - can’t switch on doubles, ranges, strings, …

- **Efficiency**
  - switch typically runs faster
  - Not a big deal
if or switch?

• Program is given a heart rate

• Needs to print out a different message depending on a range:
  – *i.e.* “Too High” if rate is > 100
  – “Normal” if rate is between 60 and 100
  – etc.

• Should we use **if** or **switch**?
Why not use switch here?

Case labels must be single constants. Consider one range:
60 through 100
How would we represent it?

case 100:
case 99:
case 98:
    . . . (the other 37 cases!)
case 60: cout << "Normal range";
Choosing If

if ( ( 0 <= heartRate ) && ( heartRate <= 200 ) )
    if ( heartRate > 100 )
        cout << "Too high";
    else if ( heartRate >= 60 )
        cout << "Normal range";
    else if ( heartRate > 0 )
        cout << "Too low";
    else
        cout << "Too late";

why not "else if ((heartRate >= 60) && (heartRate <= 100))"?
What is the output of the following code fragment? (beta is of type int.)

beta = 5;
switch (beta) {
    case 1: cout << 'R';
           break;
    case 2:
    case 4: cout << 'O';
           break;
    case 5: cout << 'L';
}
cout << 'X';

A)X  B)LX  C)RX  D)OX  E)ROLX

Ans:B
What is the output of the following code fragment? (beta is of type int.)

beta = 2;
switch (beta) {
    case 1 : cout << 'R';
            break;
    case 2 :
    case 4 : cout << 'O';
            break;
    case 5 : cout << 'L';
}
cout << 'X';  

A)X
B)LX
C)RX
D)OX
E)ROLX
Ans:D
What is the output of the following code fragment?
(beta is of type int.)

beta = 2;
switch (beta) {
    case 1 : cout << 'R';
        break;
    case 2 : cout << 'Q';
    case 4 : cout << 'O';
        break;
    case 5 : cout << 'L';
}
cout << 'X';

A)X
B)QX
C)QOX
D)QO
E)RQOLX
Ans:C
What is the output of the following code fragment? (beta is of type int.)

beta = 5;
do {
    switch (beta) {
        case 1 : cout << 'R';
            break;
        case 2 :
            case 4 : cout << 'O';
                break;
        case 5 : cout << 'L';
    }
    beta--;
} while (beta > 1);
cout << 'X';

A) X
B) ROOLX
C) LOOX
D) LOORX
E) ROOX

Ans: C
int diameter;
...
cin >> diameter;
...
if (diameter == 24)  {
    numQuarters = numQuarters + 1;
}  else if (diameter == 21)  {
    numNickels = numNickels + 1;
}  else if (diameter == 19)  {
    numPennies = numPennies + 1;
}  else if (diameter == 18 )  {
    numDimes = numDimes + 1;
}  else  {
    cout << "Not a correct diameter "
         << "for a US coin. " << endl;
}
int diameter;
...
cin >> diameter;
...
if (diameter == 24) {
    numQuarters = numQuarters + 1;
} else if (diameter == 21) {
    numNickels = numNickels + 1;
} else if (diameter == 19) {
    numPennies = numPennies + 1;
} else if (diameter == 18) {
    numDimes = numDimes + 1;
} else {
    cout << "Not a correct diameter "
         << "for a US coin. " << endl;
}

int diameter;
...
cin >> diameter;
...
switch (diameter) {
    case 24: numQuarters += 1;  break;
    case 21: numNickels  += 1;  break;
    case 19: numPennies  += 1;  break;
    case 18: numDimes    += 1;  break;
    default: cout << "Not a correct "
                << "diameter for a "
                << "US coin. " << endl;
}
```
int diameter;
...
cin >> diameter;
...
if (diameter == 24) {
    numQuarters = numQuarters + 1;
} else if (diameter == 21) {
    numNickels = numNickels + 1;
} else if (diameter == 19) {
    numPennies = numPennies + 1;
} else if (diameter == 18) {
    numDimes = numDimes + 1;
} else {
    cout << "Not a correct diameter " << "for a US coin. " << endl;
}
```

```
int diameter;
...
cin >> diameter;
...
switch (diameter) {
    case 24: numQuarters += 1; break;
    case 21: numNickels  += 1; break;
    case 19: numPennies  += 1; break;
    case 18: numDimes    += 1; break;
    default: cout << "Not a correct " << "diameter for a " << "US coin. " << endl;
}
```