The switch Statement

intro

• multi-way decision that tests an integer expression against multiple constant integer values

```java
switch (expression) {
    case const-expr: statements
    ...  // other cases
    case const-expr: statements
    default: statements
}
```
The switch Statement
semantics

switch (expression) {
    case const-expr: statements
    ...
    case const-expr: statements
    default: statements
}

• expression is evaluated to an int/char value
• execution starts at the case labeled with the int value
• execution starts at default if the int value matches none of the case labels

The switch Statement
syntax

switch (expression) {
    case const-expr: statements
    ...
    case const-expr: statements
    default: statements
}

• expression must have int (or char) type
• const-expr must be a constant:
  a literal or named constant
• statements is one or more statements
  (no braces needed)
• default is optional
The switch Statement

• Example:

```cpp
int quarter;
...
switch (quarter) {
    case 1: cout << “First”;
            break;
    case 2: cout << “Second”;
            break;
    case 3: cout << “Third”;  
            break;
    case 4: cout << “Fourth”;
            break;
    default: cout << “Invalid choice”;
}
```

The break Statement

• The break statement causes an immediate exit from the switch.

• Without a break statement, execution continues on to the next set of statements.

• Sometimes this is useful: the textbook has some nice examples.
The switch Statement

- Multiple labels for same set of statements
- if ch is ‘a’, it falls through to Option A

```cpp
char ch;
...
switch (ch) {
  case 'a':
  case 'A': cout << "Option A";
            break;
  case 'b':
  case 'B': cout << "Option B";
            break;
  case 'c':
  case 'C': cout << "Option C";
            break;
  default: cout << "Invalid choice";
}
```