Lecture 13

- Covers
  - Blocks and scope
  - The switch statement
  - The conditional operator
- Reading: Savitch 3.1

- Blocks and scope

Blocks

- A compound statement is a list of statements enclosed in { }
- A compound statement that contains variable declarations is usually referred to as a block
- Example
-
```
if (x > y)
{
    int temp = x;
    x = y;
    y = temp;
}
```

Blocks and scope

- A variable declared inside of a block is only able to be used within that block
- The scope of the variable is the part of the program in which it is usable
- A variable is not able to be accessed or set outside its scope

Blocks and scope

- What is the problem with this code?

```
{
    int number = 42;
    System.out.println(number);
}
System.out.println(number);
```

Blocks and scope

- What is output by this code?

```
int number = 42;
{
    System.out.println(number);
}
System.out.println(number);
```

Blocks and scope

- What is the problem with this code?

```
int number = 22;
{
    int number = 42;
    System.out.println(number);
}
System.out.println(number);
```

Blocks and scope

- In Java, we cannot declare variables of the same name in nested (overlapping) scope

- The switch statement

switch statement

- Many problems have several options
- Using if…else statements can get complicated
- The switch statement is a solution

- N.B. It's not always possible to replace a nested if-else statement by a switch statement

switch statement

- The switch statement compares the value of an integer or character (or expression that evaluates to an integer or character) with a number of alternatives
- Each alternative must be listed and a set of instructions specified for that alternative
- Each alternative is written inside the switch statement and is referred to as a case label

Using the switch statement

```
int month = keyboard.nextInt( );
switch (month)
{
    case 1:   System.out.println("January");
              break;
    case 2:   System.out.println("February");
              break;
    case 3:   System.out.println("March");
              break;
              …

    case 12: System.out.println("December");
              break;
    default : System.out.println("Not a valid month");
 }
```

switch general form

```
switch (controlling-expression)
{
    case constant-1:
        statement-sequence-1
        break;
    case constant-2:
        statement-sequence-2
      break;
    …

    case constant-n:
        statement-sequence-n
        break;
    default:
        default-statement-sequence
 }
```

must be of type integer or char

Default case in switch

- The default label is added to catch any values that do not match one of the case labels
- When the value of the controlling expression does not match one of the case labels, the statements executed in the switch statement start at the code following the default label

Class exercise

```
int option = keyboard.nextInt( );

switch (option)
{
    case 1:
        System.out.println("Apple");
        break;
    case 2:
        System.out.println("Banana");
        break;
    case 3:
        System.out.println("Corn");
        break;
    default:
        System.out.println("Zucchini");
}
```

- Problem
  - What is output if the input is 2?
  - What is output if the input is 0 or 4?
- Solution

The break statement

- The break statement inside a switch terminates the execution of the statement
- When a label matches the controlling expression's value, execution starts at that point and continues until a break statement is found
- When a break statement is found, the switch statement is exited (no further statements inside it are executed)

The break statement

- If a case does not contain a break statement, the processing will flow on to the next case
- One very common mistake is to accidentally omit a break statement
- But there are cases where we omit the break statement on purpose

Missed breaks

```
int option= keyboard.nextInt( );

switch (option)
{
    case 1:
        System.out.println("Apple");
    case 2:
        System.out.println("Banana");
        break;
    case 3:
        System.out.println("Corn");
        break;
    default:
        System.out.println("Zucchini");
}
```

- Accidental misses
- What is output when 1 is input?

Missed breaks

- Purposeful misses

```
case 'a':
case 'A':
    System.out.println("Excellent. You need not take the final.");
    break;
```

- If no break statement is found, processing continues onto the next case

Example

- Display message according to

Grade	Message
A	*You need not take the exam*
B	*Your grade is now A*
C	*Passing*
D, F	*Not good - more study needed!*
others	*Invalid grade*

Solution

```
String gradeString = keyboard.nextLine();
char grade = gradeString.charAt(0);
switch (grade)
{
    case 'A': System.out.println("You need not take the exam");
              break;
    case 'B': grade = 'A';
              System.out.println("Your grade is now " + grade);
              break;
    case 'C': System.out.println("Passing");
              break;
    case 'D':
    case 'F': System.out.println("Not good - more study needed!");
              break;
    default : System.out.println("Invalid grade");
 }
```

Example

- Write a switch statement that checks the value of an integer variable month and depending on the month, outputs the number of days in it.
- Values of month are interpreted as:

  1 = January, 2 = February, …, 12 = December

```
switch (month)
{
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        System.out.println("31 Days");
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        System.out.println("30 Days");
        break;
    case 2:
        System.out.println("28 days (29 days in a leap year)");
        break;
    default :
        System.out.println("Not a valid month");
 }
```

Class exercise

- Write a switch statement that displays whether or not you should go to university depending on the day of the week entered
  - Assume you attend Monday to Friday
  - Assume 1 = Sunday, 2 = Monday, etc.
  - Cater for a number that is not between 1 and 7 with an error message

Solution

Class exercise

- The integer variable mark stores a student's final mark
- Write a switch statement that outputs
  - 'A' if the mark is between 80 & 100 (inclusive)
  - 'B' if the mark is between 70 & 79 (inclusive)
  - 'C' if the mark is between 60 & 69 (inclusive)
  - 'D' if the mark is between 50 & 59 (inclusive)
  - 'F' otherwise

Solution

Limits of switch

- NOTE: Not all multibranch if-else statements can be replaced by a switch statement

- Conditional operator

The conditional operator

- The conditional operator is an older style of branching statement
- It tests a boolean expression and depending on the truth value of that expression, returns one of two specified values
- It is a ternary operator (i.e. requires 3 operands)

```
max = (n1 > n2) ? n1 : n2
```

Example

```
String motto = optimist ? "The glass is half full"
                        : "The glass is half empty";

System.out.println("You " +
        ((height > 150) && (age > 16)? "can" : "can't") +
        " go on the ride.");
```

- Prefer the if-else statement to the conditional operator
- All but the simplest statements using the conditional operator can be difficult to understand

Class exercise

- Write a statement that sets the value of the integer variable absolute to the absolute value of the variable n
- Use the conditional operator in your statement

Solution

Next lecture

- Looping statements
  - The while statement
  - The do…while statement
  - Infinite loops