switch

a decision structure where one choice, of possibly many, different choices are made

```
switch ( expression ) {
case value1 : statement list1
case value2 : statement list2
case value3 : statement list3
...
default : default statement list
}
```

According to the **value** of the **expression** a **case** is executed

Execution of statements follows unless a **break** statement is encountered

The type of the expression must be char, byte, short, int, Character, Byte, Short, Integer, String, or an enum

Example 1

translating a letter grade to a numeric grade

letter grade	grade point
A	4
B	3
C	2
D	1
F	0

Example 1

```
String grade;
double nGrade;
System.out.println("Enter letter grade:");
Scanner kb = new Scanner(System.in);
grade = kb.next();
switch (grade) {
    case "A": nGrade = 4.0;
              break;
    case "B": nGrade = 3.0;
              break;
    case "C": nGrade = 2.0;
              break;
    case "D": nGrade = 1.0;
              break;
    default: nGrade = 0.0;
}
System.out.println(grade+" --> "+nGrade);
```

expression

break

If expression != any of the above cases values

Conditional operator ? :

A decision structure that can replace an if-then-else

```
if ( a > b ) larger = a
else larger = b;
```

Can be written using ?: as

```
larger = ( a > b ) ? a : b;
```

General recommendation: use it when it makes your code clearer.

You will see **?:** used frequently.

Conditional operator ? : (not in text)

A decision structure that can replace an if-then-else

```
if ( a > b ) System.out.println(a);
else System.out.println(b);
```

Can be written using ?: as

```
System.out.println( ( a > b ) ? a : b);
```

You can use ?: anywhere an expression could appear.

Conditional operator ? :

General recommendation: use it when it makes your code clearer.

Maybe these are not so clear:

```
for (int i=0; i<line.length(); i++){
     char c = line.charAt(i);
     encrypted+= (c=='a')?'e':((c=='e')?'a':c);
}
```

Or

```
      encrypted+= c=='a'?'e':c=='e'?'a':c;
```