Class Declaration

```
public class CashRegister
{
   private int itemCount;
   private double totalPrice;

   public void addItem(double price)
   {
      itemCount++;
      totalPrice = totalPrice + price;
   }
   . . .
}
```

Instance variables

Method

Selected Operators and Their Precedence

(See Appendix B for the complete list.)

`[]`	Array element access
`++ -- !`	Increment, decrement, Boolean *not*
`* / %`	Multiplication, division, remainder
`+ -`	Addition, subtraction
`< <= > >=`	Comparisons
`== !=`	Equal, not equal
`&&`	Boolean *and*
`||`	Boolean *or*
`=`	Assignment

Conditional Statement

```
if (floor >= 13)
{
   actualFloor = floor - 1;
}
else if (floor >= 0)
{
   actualFloor = floor;
}
else
{
   System.out.println("Floor negative");
}
```

Condition

Executed when condition is true

Second condition (optional)

Executed when all conditions are false (optional)

Variable and Constant Declarations

```
int cansPerPack = 6;

final double CAN_VOLUME = 0.335;
```

Type

Name

Initial value

Method Declaration

```
public static double cubeVolume(double sideLength)
{
   double volume = sideLength * sideLength * sideLength;
   return volume;
}
```

Modifiers

Return type

Parameter type and name

Exits method and returns result.

Mathematical Operations

`Math.pow(x, y)`	Raising to a power x^y
`Math.sqrt(x)`	Square root $\sqrt{x}$
`Math.log10(x)`	Decimal log $\log_{10}(x)$
`Math.abs(x)`	Absolute value $\lvert x \rvert$

`Math.sin(x)`
`Math.cos(x)`
`Math.tan(x)`
	Sine, cosine, tangent of x (x in radians)

String Operations

```
String s = "Hello";
int n = s.length(); // 5
char ch = s.charAt(1); // 'e'
String t = s.substring(1, 4); // "ell"
String u = s.toUpperCase(); // "HELLO"
if (u.equals("HELLO")) ... // Use equals, not ==
for (int i = 0; i < s.length(); i++)
{
   char ch = s.charAt(i);
   Process ch
}
```

Loop Statements

```
while (balance < TARGET)
{
   year++;
   balance = balance * (1 + rate / 100);
}
```

Condition

Executed while condition is true

```
for (int i = 0; i < 10; i++)
{
   System.out.println(i);
}
```

Initialization

Condition

Update

```
do
{
   System.out.print("Enter a positive integer: ");
   input = in.nextInt();
}
while (input <= 0);
```

Loop body executed at least once

```
for (double value : values)
{
   sum = sum + value;
}
```

Set to a new element in each iteration

An array or collection

Executed for each element