Review
Introduction to Computer Science I
CSE 1020

moodle.yorku.ca

Early binding

When the compiler encounters the invocation

$\mathrm{C.m}(a_1, \ldots, a_n)$

it must determine which method to invoke. This process is known as early binding. It consists of the following three steps.

1. Find the class C.
2. Find compatible methods m in class C.
3. Select the most specific compatible method m in class C.

Early binding

Question

Which methods in class $\mathrm{PrintStream}$ are compatible with invocation $\mathrm{output.println}(1)$?

Early binding

Question

Which methods in class $\mathrm{PrintStream}$ are compatible with invocation $\mathrm{output.println}(1)$?

Answer

$\mathrm{println}(\mathrm{double})$
$\mathrm{println}(\,\mathrm{float}\,)$
$\mathrm{println}(\,\mathrm{int}\,)$
$\mathrm{println}(\mathrm{long})$

Early binding

Question

Which of the methods

$\mathrm{println}(\mathrm{double})$
$\mathrm{println}(\,\mathrm{float}\,)$
$\mathrm{println}(\,\mathrm{int}\,)$
$\mathrm{println}(\mathrm{long})$

in class $\mathrm{PrintStream}$ is most specific to invocation $\mathrm{output.println}(1)$?

Early binding

Question

Which of the methods

$\mathrm{println}(\mathrm{double})$
$\mathrm{println}(\mathrm{float})$
$\mathrm{println}(\mathrm{int})$
$\mathrm{println}(\mathrm{long})$

in class $\mathrm{PrintStream}$ is most specific to invocation $\mathrm{output.println}(1)$?

Answer

$\mathrm{println}(\mathrm{int})$ since the argument 1 is of type int.

Early binding

Question

Which of the methods

$\mathrm{println}(\mathrm{double})$
$\mathrm{println}(\,\mathrm{float}\,)$
$\mathrm{println}(\,\mathrm{int}\,)$
$\mathrm{println}(\mathrm{long})$

in class $\mathrm{PrintStream}$ is most specific to invocation
$\mathrm{output.println}(1\mathrm{L})$?

Early binding

Question

Which of the methods

$\mathrm{println}(\mathrm{double})$
$\mathrm{println}(\,\mathrm{float}\,)$
$\mathrm{println}(\,\mathrm{int}\,)$
$\mathrm{println}(\mathrm{long})$

in class $\mathrm{PrintStream}$ is most specific to invocation $\mathrm{output.println}(1\mathrm{L})$?

Answer

$\mathrm{println}(\mathrm{long})$ since the argument $1\mathrm{L}$ is of type long.

Early binding

Question

Which of the methods

```
println(double)
println( float )
println( int )
println(long)
```

in class PrintStream is most specific to invocation output.println ('1') ?

Early binding

Question

Which of the methods

$\mathrm{println}(\mathrm{double})$
$\mathrm{println}(\mathrm{float})$
$\mathrm{println}(\mathrm{int})$
$\mathrm{println}(\mathrm{long})$

in class $\mathrm{PrintStream}$ is most specific to invocation $\mathrm{output.println}$ ('1') ?

Answer

$\mathrm{println}(\mathrm{int})$ since the argument '1' is of type char and converting it to an int requires the least amount of promotion.

What is a class?

A class contains (non-static) attributes. Each attribute has a name and a type.

numerator : long
denominator : long

What is a class?

- What is the numerator of this fraction?
- What is the denominator of this fraction?
- ...

A class contains (non-static) methods. Each method has a signature and possibly a return type.

getNumerator() : long
getDenominator() : long

What is an object?

An object is an instance of a class.

An object has a state. The state of an object consists of the non-static attributes of the class and their values.

numerator	1
denominator	7

What is an object?

An object has an identity. This identity is unique. That is, two different objects have different identities.

This is an abstract notion. In more concrete terms, you may think of an object's identity as the address in memory where it is stored. Obviously, two different objects cannot be stored at the same memory address.

What is a class?

A class contains constructors. Each constructor has a signature, name of which is the same as the name of the class.

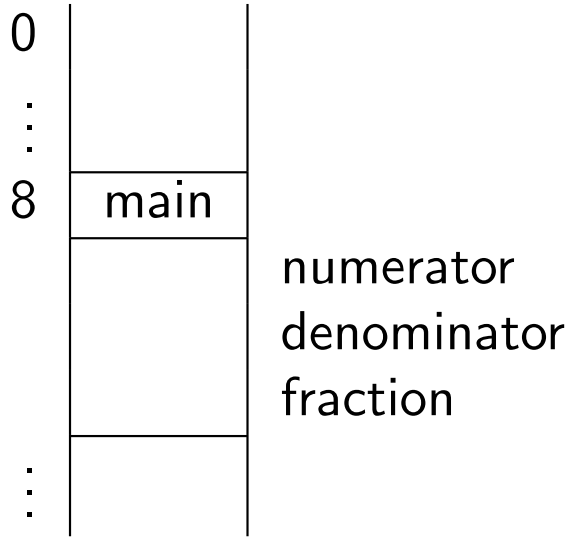
Fraction()
Fraction(long, long)

How to create objects?

```
output.print("Enter the numerator: ");
long numerator = input.nextLong();
output.print("Enter the denominator: ");
long denominator = input.nextLong();
Fraction fraction = new Fraction(numerator, denominator);
```

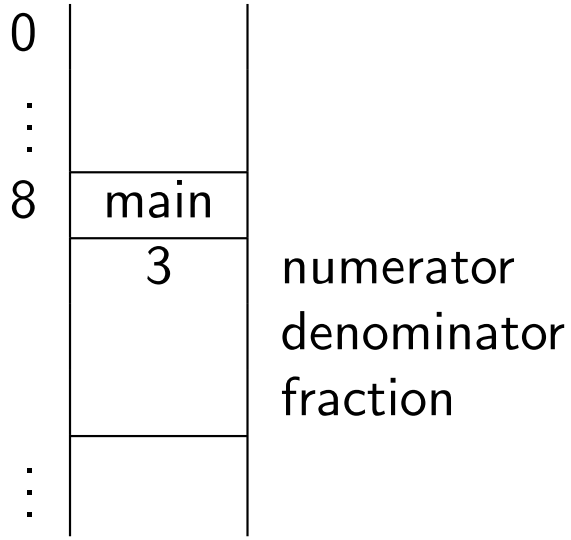
How to create objects?

```
long numerator = 3;
long denominator = 4;
Fraction fraction = new Fraction(numerator, denominator);
```



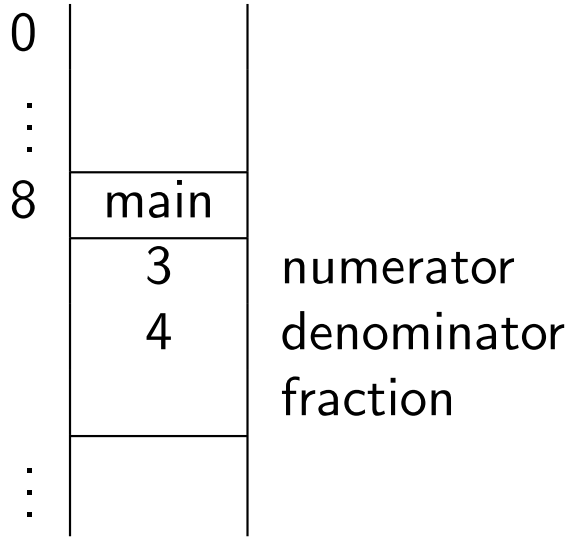
How to create objects?

```
long numerator = 3;
long denominator = 4;
Fraction fraction = new Fraction(numerator, denominator);
```



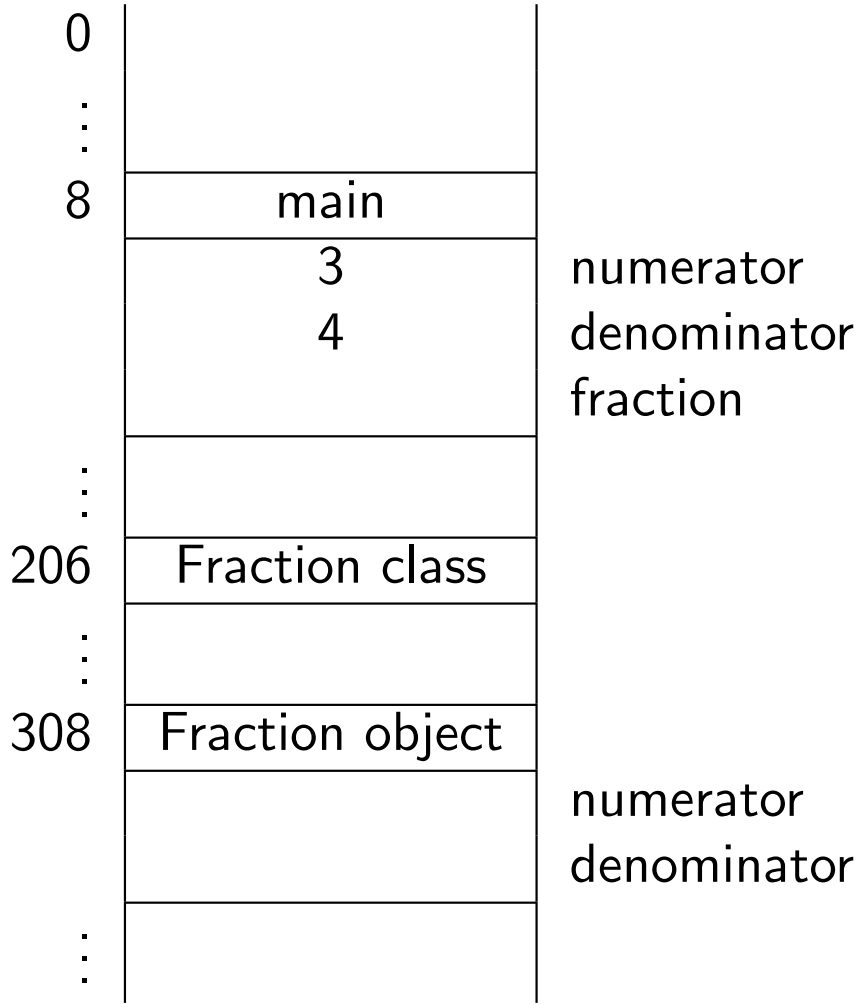
How to create objects?

```
long numerator = 3;
long denominator = 4;
Fraction fraction = new Fraction(numerator, denominator);
```



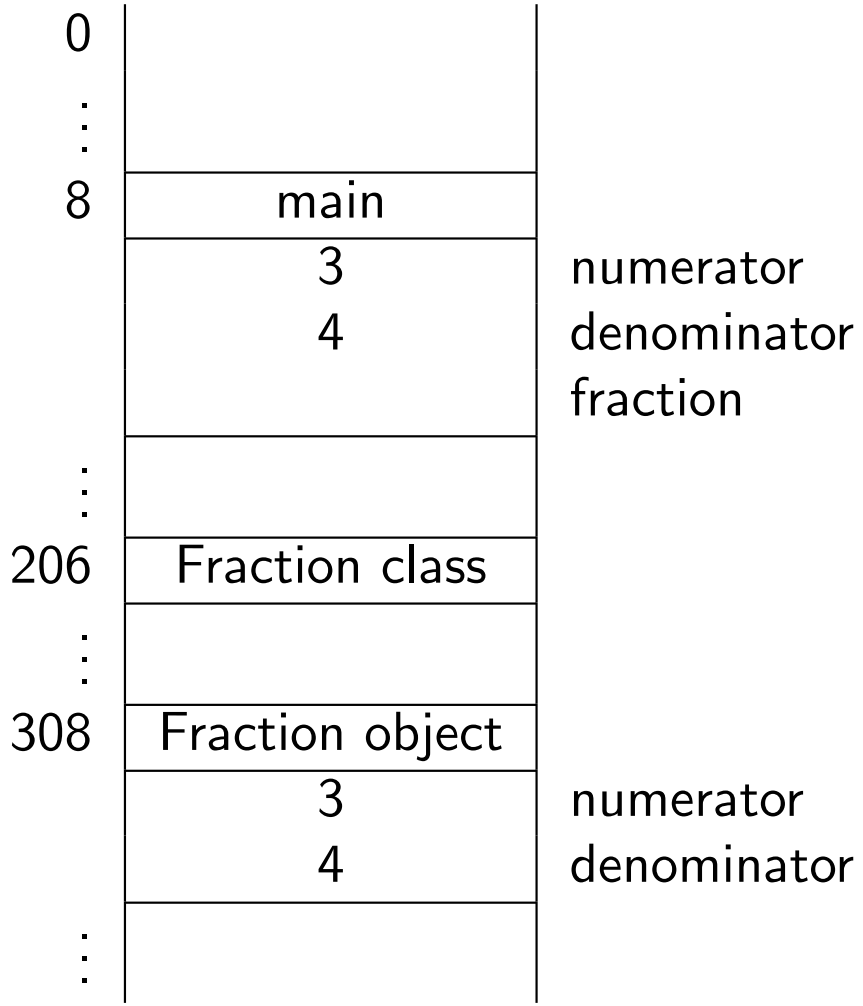
How to create objects?

```
long numerator = 3;
long denominator = 4;
Fraction fraction = new Fraction(numerator, denominator);
```



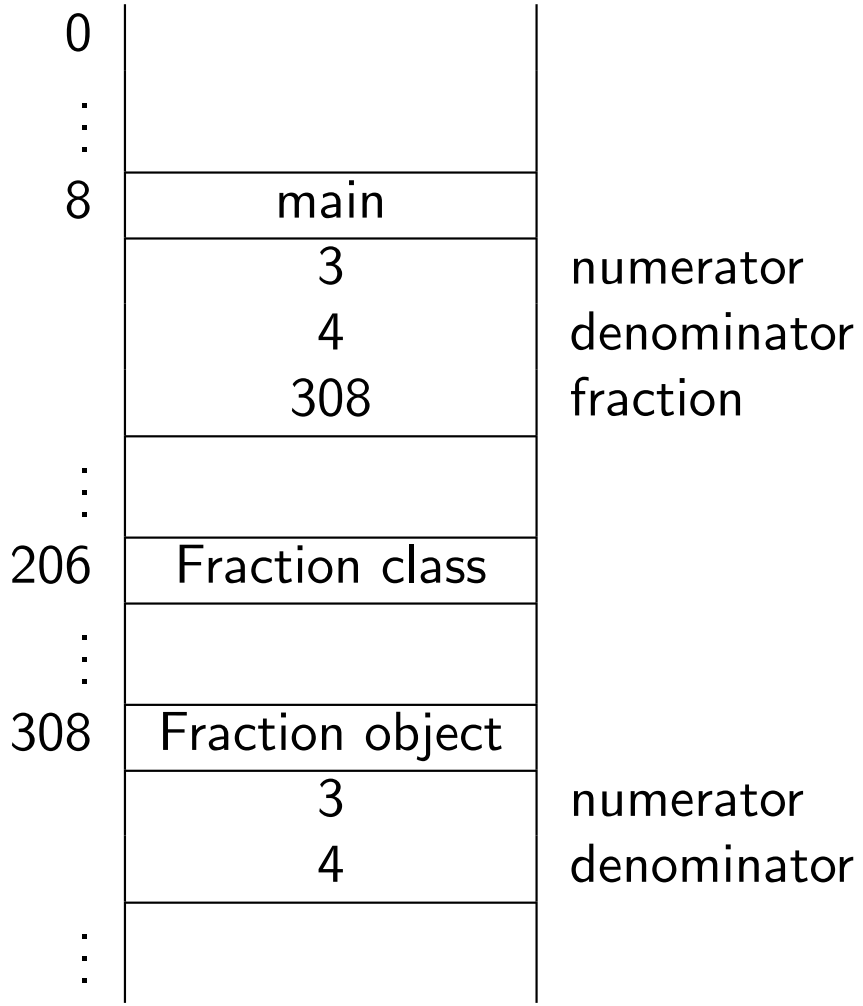
How to create objects?

```
long numerator = 3;
long denominator = 4;
Fraction fraction = new Fraction(numerator, denominator);
```



How to create objects?

```
long numerator = 3;
long denominator = 4;
Fraction fraction = new Fraction(numerator, denominator);
```



Invoking a non-static method

The invocation

```
first.add(second)
```

contains two object references:

- `first` is (a reference to) the object on which the method is invoked, and
- `second` is (a reference to) the object that is provided as an argument to the method.

Invoking a non-static method

⋮		
308	Fraction object	
	1	numerator
	7	denominator
⋮		
408	Fraction object	
	1	numerator
	7	denominator
⋮		
452	add	
	308	this
	408	other
⋮		

Invoking a non-static method

Question

Does the method

```
public void add(Fraction other)
```

return anything?

Invoking a non-static method

Question

Does the method

```
public void add(Fraction other)
```

return anything?

Answer

No.

Invoking a non-static method

Question

Does the method

```
public void add(Fraction other)
```

return anything?

Answer

No.

Question

If it does not return anything, does it do anything?

Invoking a non-static method

Question

Does the method

```
public void add(Fraction other)
```

return anything?

Answer

No.

Question

If it does not return anything, does it do anything?

Answer

Yes, it changes the state of the object on which it is invoked.

Garbage collection

```
Fraction f = new Fraction();
Fraction g = new Fraction(1, 2);
Fraction h = new Fraction();
f = g;
h = null;
```

Draw the diagram representing the memory once the execution has reached the end of the snippet.

Garbage collection

100	main	
	400	f
	400	g
	null	h
200	Fraction class	
300	Fraction object	
	0	numerator
	1	denominator
400	Fraction object	
	1	numerator
	2	denominator
500	Fraction object	
	0	numerator
	1	denominator

Question

How many object references refer to the objects at addresses 300 and 500?

Garbage collection

Question

How many object references refer to the objects at addresses 300 and 500?

Answer

Zero.

The objects at addresses 300 and 500 have become orphans.

Every now and then, the garbage collector removes all orphans from memory.

Passing of arguments

Question

What is the difference between pass-by-value and pass-by-reference?

Passing of arguments

Question

What is the difference between pass-by-value and pass-by-reference?

Answer

In pass-by-value, the values of the arguments are passed, whereas in pass-by-reference, the addresses of the arguments are passed.

Pass-by-value

Question

What is the output produced by the following code snippet?

```
int x = 0;
int y = 1;
Magic.swap(x, y);
output.println(x);
output.println(y);
```

Pass-by-value

Question

What is the output produced by the following code snippet?

```
int x = 0;
int y = 1;
Magic.swap(x, y);
output.println(x);
output.println(y);
```

Answer

```
0
1
```

Pass-by-value or pass-by-reference?

Question

The code snippet

```
Fraction f = new Fraction(0, 1);
Fraction g = new Fraction(1, 1);
Magic.swap(f, g);
output.println(f);
output.println(g);
```

produces the output

```
1/1
0/1
```

Can this output be a result of pass-by-value?

Pass-by-value or pass-by-reference?

Question

The code snippet

```
Fraction f = new Fraction(0, 1);
Fraction g = new Fraction(1, 1);
Magic.swap(f, g);
output.println(f);
output.println(g);
```

produces the output

```
1/1
0/1
```

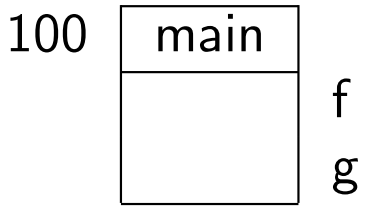
Can this output be a result of pass-by-value?

Answer

Yes!

Pass-by-value

```
Fraction f = new Fraction(0, 1);
Fraction g = new Fraction(1, 1);
Magic.swap(f, g);
output.println(f);
output.println(g);
```



Pass-by-value

```
Fraction f = new Fraction(0, 1);
Fraction g = new Fraction(1, 1);
Magic.swap(f, g);
output.println(f);
output.println(g);
```

100	main	
	300	f
	400	g
200	Fraction class	
300	Fraction object	
	0	numerator
	1	denominator
400	Fraction object	
	1	numerator
	1	denominator

Pass-by-value

```
Fraction f = new Fraction(0, 1);
Fraction g = new Fraction(1, 1);
Magic.swap(f, g);
output.println(f);
output.println(g);
```

100	main	
	300	f
	400	g
200	Fraction class	
300	Fraction object	
	0	numerator
	1	denominator
400	Fraction object	
	1	numerator
	1	denominator
500	Magic.swap	
	300	first
	400	second

Pass-by-value

```
Fraction f = new Fraction(0, 1);
Fraction g = new Fraction(1, 1);
Magic.swap(f, g);
output.println(f);
output.println(g);
```

100	main	
	300	f
	400	g
200	Fraction class	
300	Fraction object	
	1	numerator
	1	denominator
400	Fraction object	
	0	numerator
	1	denominator
500	Magic.swap	
	300	first
	400	second

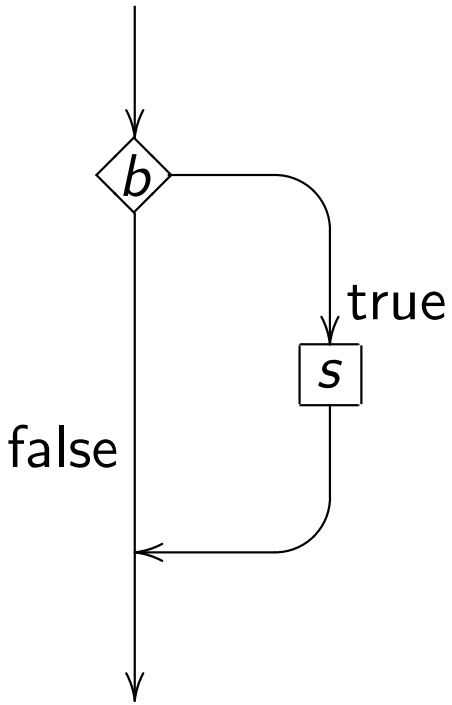
Pass-by-value

Note that

- the values of f and g are not modified (just like the values of x and y were not modified either),
- but the states of the objects to which f and g refer are modified.

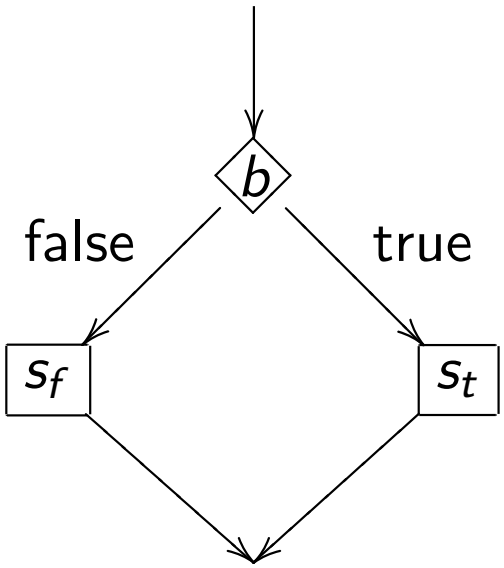
If statement

```
if (b) {
    s
}
```



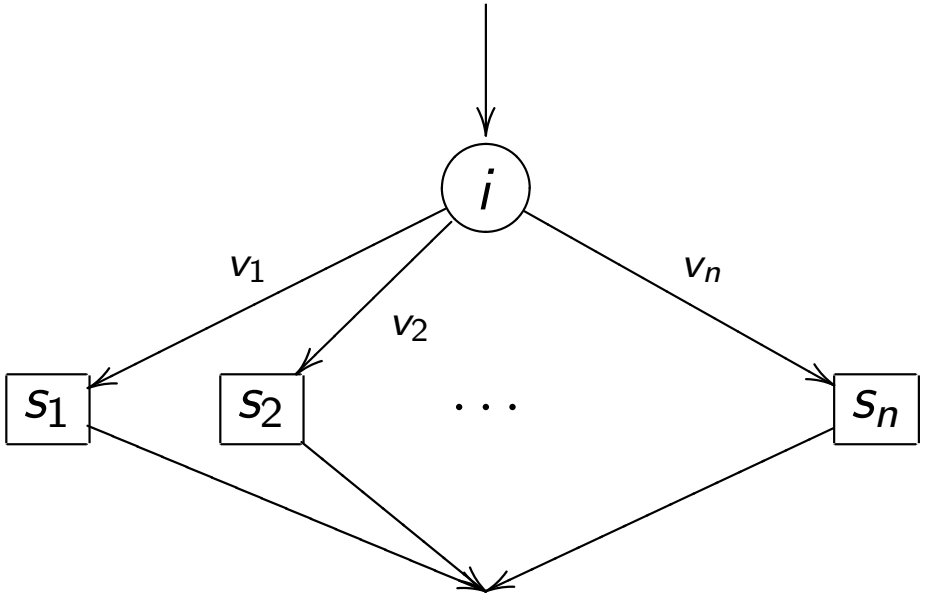
If-else statement

```
if (b) {
    st
} else {
    sf
}
```



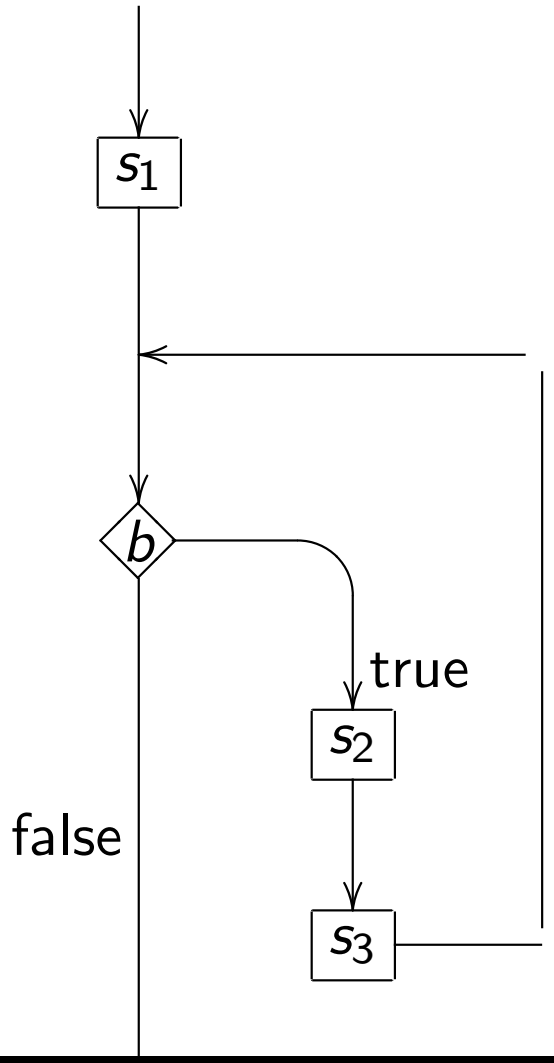
Switch statement

```
switch (i) {
  case v1 : s1
            break;
  case v2 : s2
            break;
  ...
  case vn : sn
            break;
}
```



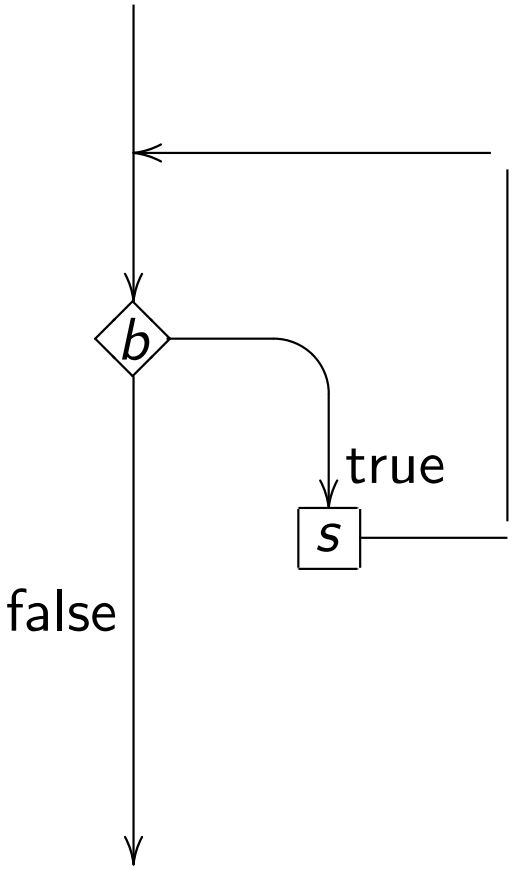
For statement

```
for(s1; b; s3) {
  s2
}
```



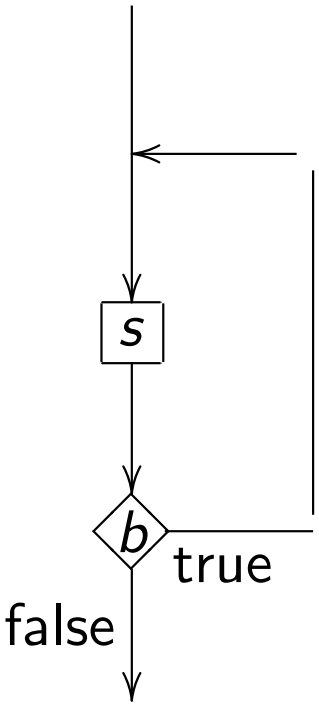
While statement

```
while (b) {
  s
}
```



Do statement

```
do {
  s
} while (b);
```



Check04B

The `Check04B` app prompts for a stock symbol and outputs the current date and time, the name of the stock, and its price. Here is a sample run:

```
Enter a stock symbol ... .sx
As of Thu Feb 6 15:24:11 EDT 2014, the price of a
Compu-SIERRA & X_RAY Corp.
share is: 35.86
```

The first line contains the prompt and the user's input (you may assume that the entered symbol is valid). The second line contains the words "as of" followed by the date and time (and time zone) at which the program was run, followed by ", the price of a". The third line contains the name of the stock as listed on the exchange. The fourth line contains the words "share is:" followed by the price rounded to two decimals.

Check05D

Given a real number a, we want to compute the limit of the sequence:

$$x_{new} = x - (x^3 - a)/3x^2$$

where the initial value for x is a. In other words, you start by setting $x = a$ and computing the right-hand side. The result will be the next value of x. Repeat this process until you reach a stage at which the new and the old values of x are the same (within some tolerance). This final value is the limit of the sequence. Create the app `Check05D` that uses 0.001 as tolerance, reads a, and outputs the limit. The app stops iterating when the absolute value of the difference between the old and new values of x become less than the tolerance. Here is a sample run of the sought app:

```
Enter any real number ... 500
Within 0.0010 the limit is: 7.9370
```

Note that the tolerance and the limit are formatted using four decimals.