Static Attributes

Question

How do you use a static attribute `a` of a class `C`?

Static Attributes

Question

How do you use a static attribute a of a class `C`?

Answer

`C.a`

Static Attributes

Question

How do you use a static attribute a of a class `C`?

Answer

`C.a`

Question

How do you print the value of the static attribute `PI` of the class `Math`?

Static Attributes

Question

How do you use a static attribute `a` of a class `C`?

Answer

```
C.a
```

Question

How do you print the value of the static attribute `PI` of the class `Math`?

Answer

```
output.println(Math.PI);
```

Static Methods

Question

How do you use a static method `m` of a class `C`?

Static Methods

Question

How do you use a static method `m` of a class `C`?

Answer

`C.m(...)`

Static Methods

Question

How do you use a static method `m` of a class `C`?

Answer

`C.m(...)`

Question

How do you print the value of $2.0^{10.0}$ using the static method `pow` of the class `Math`?

Static Methods

Question

How do you use a static method `m` of a class `C`?

Answer

```
C.m(...)
```

Question

How do you print the value of $2.0^{10.0}$ using the static method `pow` of the class `Math`?

Answer

```
final double BASE = 2.0;
final double EXPONENT = 10.0;
output.println(Math.pow(BASE, EXPONENT));
```

Non-Static Methods

Question

How do you use a non-static method `m` of a class `C`?

Non-Static Methods

Question

How do you use a non-static method `m` of a class `C`?

Answer

`o.m(...)` where `o` is an object reference of type `C`.

Non-Static Methods

Question

How do you use a non-static method `m` of a class `C`?

Answer

`o.m(...)` where `o` is an object reference of type `C`.

Question

How do you print the value of $\frac{2}{3}$ using the non-static method `toString` of the class `Fraction`?

Non-Static Methods

Question

How do you use a non-static method `m` of a class `C`?

Answer

`o.m(...)` where `o` is an object reference of type `C`.

Question

How do you print the value of $\frac{2}{3}$ using the non-static method `toString` of the class `Fraction`?

Answer

```
Fraction fraction = new Fraction(2, 3);
output.println(fraction.toString());
```

Static Attributes

Question

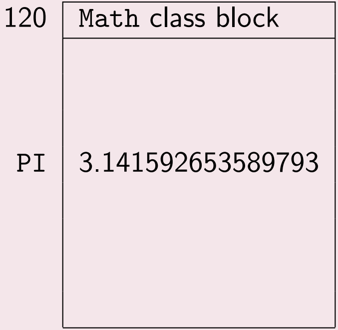
Where in our memory diagrams do you find the static attribute `PI` of the class `Math`?

Static Attributes

Question

Where in our memory diagrams do you find the static attribute `PI` of the class `Math`?

Answer



Only the `Math` class has the `PI` attribute.

Non-Static Attributes

Question

Where in our memory diagrams do you find the non-static attribute `numerator` of the class `Fraction`?

Question

Where in our memory diagrams do you find the non-static attribute `numerator` of the class `Fraction`?

Answer

230	Fraction object block
`numerator`	2
`denominator`	3
410	Fraction object block
`numerator`	1
`denominator`	4

Each `Fraction` object has a `numerator` attribute.

Check03A

Number of students enrolled in the course: 225

Number of students that eChecked Check03A: 70 (31%)

Check04D

Number of students enrolled in the course: 215

Number of students that eChecked Check03A: 46 (21%)

Programming Test 2

Number of students that submitted a program: 202

Number of programs that did not compile: 77 (38%)

If the program you submit does not compile, then the maximal mark for that program is 4 (out of 10).

Structure of our apps

```
public class ...
{
  public static void main(String[] arguments)
  {
     ...
  }
}
```

Ingredients of the `main` Method

Question

Which "instructions" do we use in the `main` method?

Ingredients of the `main` Method

Question

Which "instructions" do we use in the `main` method?

Answer

- declarations

  `type variable;`
- assignments

  `variable = expression;`
- method invocations

  `Class.method(arguments);` and

  `object.method(arguments);`

Many problems cannot be solved using only the above "instructions."

Control Structures

CSE 1020

October 6, 2010

Control Structures

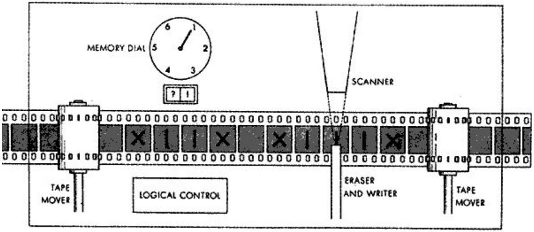
- if statement
- if-else statement
- switch statement
- for statement
- while statement
- do statement

Any of the last three control structures makes Java a so-called Turing complete language.

Turing completeness

Definition

A programming language is *Turing complete* if a simulator of a Turing machine can written in the programming language.



This notion will be covered in more detail in the course "Introduction to the Theory of Computation" (CSE 2001).

Alan Turing

Alan Turing (June 23, 1912–June 7, 1954) was an English mathematician. He formalized the notion of computation by means of a machine. This machine was later named the Turing machine. The Turing award, the "Nobel prize of computing" is named after him.

source: ieee.org

Problem5_1

Problem

Prompt the user for their percentage grade by printing

```
Enter your mark (0.0-100.00):
```

so that the grade is entered by the user on the same line as the prompt. On the next line, print

```
Passed?
```

followed by `true` if the grade is greater than or equal to 50.0, and `false` otherwise.

Problem5_2

Problem

Prompt the user for their percentage grade by printing

```
Enter your mark (0.0-100.00):
```

so that the grade is entered by the user on the same line as the prompt. On the next line, print

```
Passed
```

if the grade is greater than or equal to 50.0, and

```
Failed
```

otherwise.

Problem5_3

Problem

Prompt the user for their percentage grade by printing

```
Enter your mark (0.0-100.00):
```

so that the grade is entered by the user on the same line as the prompt. On the next line, print the corresponding letter grade (A, B, C, D, E or F).

Problem5_4

Problem

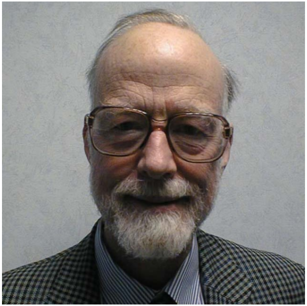
Prompt the user for their letter grade by printing

```
Enter your mark (A-F):
```

so that the grade is entered by the user on the same line as the prompt. On the next line, print the corresponding percentage grade (0.0-100.0).

Tony Hoare

Sir Charles Antony Richard Hoare (born January 11, 1934) is a British computer scientist. He is best known for the development of Quicksort, an algorithm to sort elements. He also proposed the switch statement. In 1980, he received the Turing award.



source: research.microsoft.com

Problem5_5

Problem

Prompt the user for a non-negative integer

```
Enter a non-negative integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print n $*$'s.

Loop Invariant

Definition

Given a loop, a boolean expression is a *loop invariant* of the loop if it holds at the beginning of every iteration of the loop.

C.A.R. Hoare. An Axiomatic Basis for Computer Programming. *Communications of the ACM*, 12(10): 576–580, October 1969.

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++)
{
  output.print("*");
}
```

Loop invariants for this loop are

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++)
{
  output.print("*");
}
```

Loop invariants for this loop are

- true

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++)
{
  output.print("*");
}
```

Loop invariants for this loop are

- true
- $\texttt{i} \geq 0$

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++)
{
  output.print("*");
}
```

Loop invariants for this loop are

- true
- $\texttt{i} \geq 0$
- $\texttt{i} \leq n$

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++)
{
  output.print("*");
}
```

Loop invariants for this loop are

- true
- i ≥ 0
- i $\leq n$
- i $*$'s have been printed

Loop Invariant

Consider the loop

```
for (int i = 0; i < n; i++)
{
  output.print("*");
}
```

Loop invariants for this loop are

- true
- $\texttt{i} \geq 0$
- $\texttt{i} \leq n$
- i $*$'s have been printed
- $\texttt{i} \geq 0$ && $\texttt{i} \leq n$ && i $*$'s have been printed

Problem

Prompt the user for a non-negative integer

```
Enter a non-negative integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print 1, 2, $\ldots n - 1$, n, separated by a single space.

Problem5_7

Problem

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompt. On the next line, print

n `is prime`

if n is prime and

n `is not prime`

otherwise.

Problem5_8

Problem

Prompt the user for a file name

```
Enter a file name:
```

so that the name is entered by the user on the same line as the prompt. Print the content of the file.

For and While Loops

Theorem

Every for loop can be expressed as a while loop.

Proof.

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

can be expressed as

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

Theorem

Every while loop can be expressed as a for loop.

Problem5_9

Problem

Prompt the user for two positive integers

```
Enter the number of rows:
Enter the number of columns:
```

so that the integers r and c are entered by the user on the same line as the prompts. Print r lines each consisting of c $*$'s.

Problem5_10

Problem

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, ..., a line with $n-1$ $*$'s, and a line with n $*$'s.

Problem

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Prompt the user for a file name

```
Enter a file name:
```

so that the name is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, $\ldots$, a line with $n-1$ $*$'s, and a line with n $*$'s to the given file.

Problem5_12

Problem

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, ..., a line with $n-1$ $*$'s, and a line with n $*$'s. Crash if the user enters a non-positive integer.

Problem5_13

Problem

Prompt the user for a positive integer

```
Enter a positive integer:
```

so that the integer n is entered by the user on the same line as the prompts. Print a line with 1 $*$, a line with 2 $*$'s, ..., a line with $n-1$ $*$'s, and a line with n $*$'s. Reprompt the user if they enter a non-positive integer.

For and Do Loops

Theorem

Every for loop can be expressed as a do loop.

Theorem

Every do loop can be expressed as a for loop.

Question

So which loop should we use?

For and Do Loops

Theorem

Every for loop can be expressed as a do loop.

Theorem

Every do loop can be expressed as a for loop.

Question

So which loop should we use?

Answer

It is a matter of taste. If you know the number of iterations in advance, a for loop may be most appropriate. If the loop has to be executed at least once, a do loop may be most appropriate.