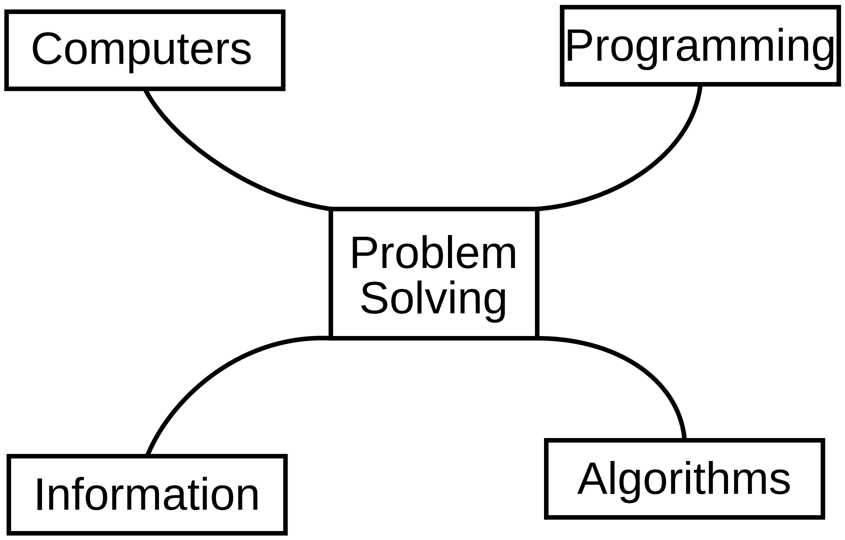
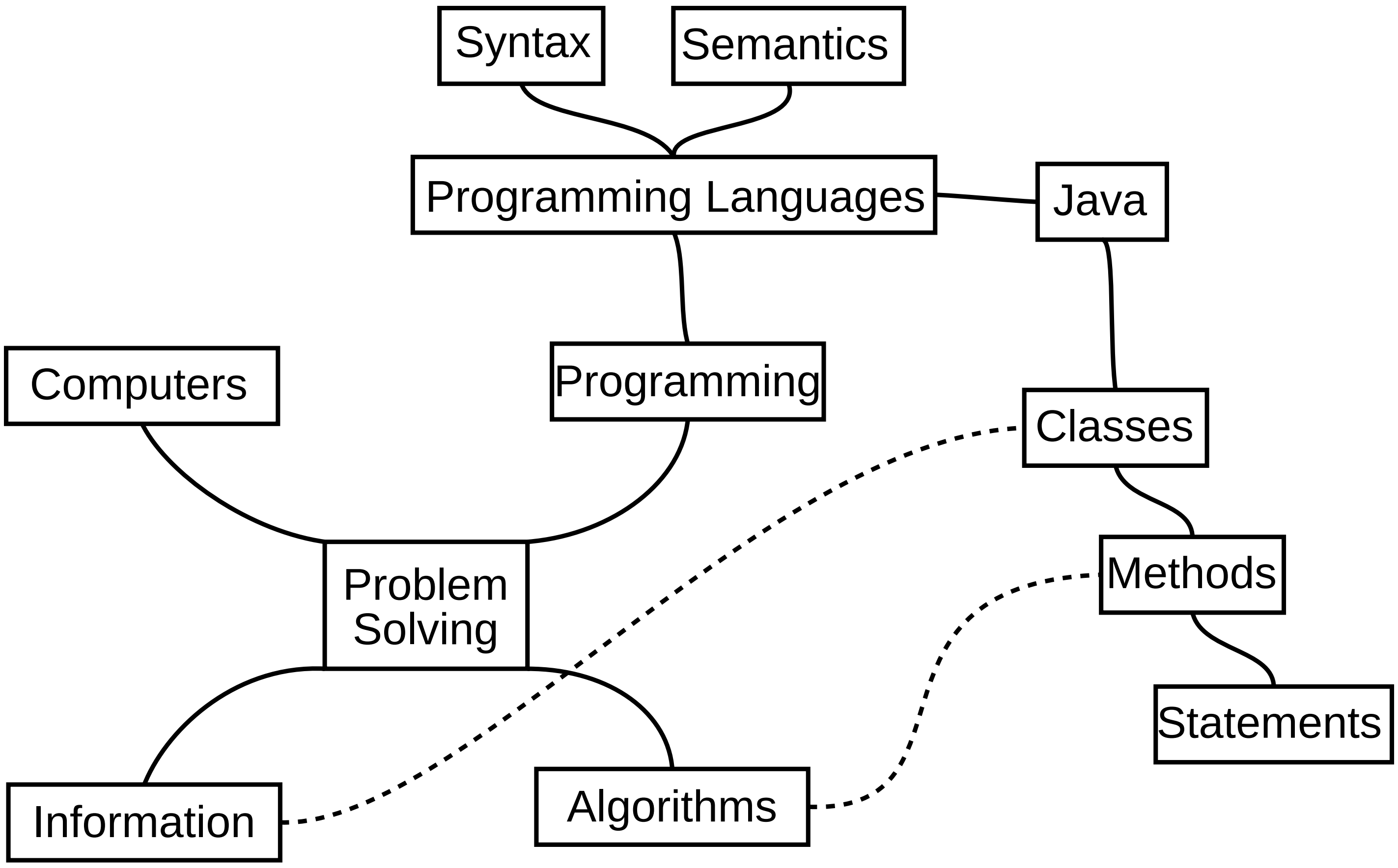
Announcements

- Assignment 3, posted today. Suggestion: start soon!

Road map


Computers
Programming
Problem Solving
Information
Algorithms

Road map

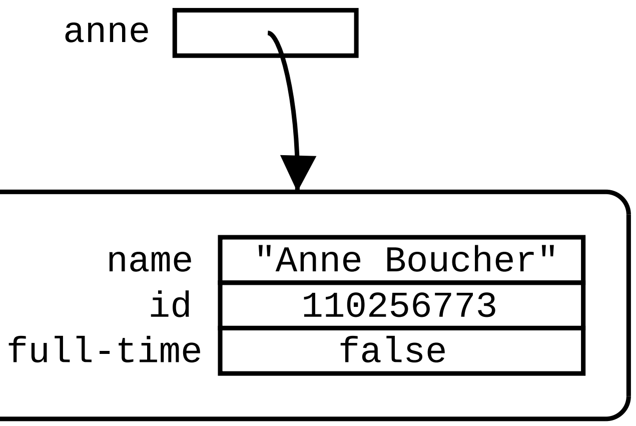

Syntax
Semantics
Programming Languages
Java
Computers
Programming
Classes
Problem Solving
Methods
Statements
Information
Algorithms

Statements

- Variable declaration
```
   type variable;
```
- Assignment
```
   variable = expression;
```
- Method invocation
```
   objectreference.methodname(parameters);
```
  or
```
   classname.methodname(parameters);
```
- Conditional
```
   if (condition) block;
```
  or
```
   if (condition) block1; else block2;
```
- Loop
```
   while (condition) block;
```

Objects and Classes

- Programs manipulate data
- Variables store data
- A variable holds either:
  - a value from a primitive data type (int, boolean, char, ...)
  - or a reference to an object
- An *object* is a composite piece of data: it is a group of variables treated as a unit



Objects and classes

- The data type of an object is a *class*
- Classes have *methods*
- Methods are the operations of a class
- Applying a method to an object is written:

  ```
  objectreference.methodname(parameters)
  ```

  where `methodname` is defined in the class of the object
- For example:

  ```
  anne.change_id(260298776)
  ```

Class definition

```
public class Name
{
    // Attribute definitions
    // ...

    // Method definitions
    // ...
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    // Methods
    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
    // ...
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    // Methods
    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
    // ...
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    // Methods
    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
    // ...
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    // Methods
    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
    // ...
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    // Methods
    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
    // ...
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    // Methods
    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
    // ...
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    while (!radio_on) current_song++;  //WRONG!

    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
}
```

Objects and classes

- A class is a "type" of objects. Objects are the values of a class.
- A class is defined by the attributes shared by all its objects, and by its methods
- The attributes of a class represent those characteristics which all objects of the class *have*: e.g. every student has a *name* and an *id*. Hence, `name` and `id` can be attributes of a `Student` class.
- The methods of a class represent the operations that can be performed on objects of that class, they define how an object in the class reacts to "messages" sent to it by other objects: e.g. the method `play` in the `Stereo` class, defines how all stereo objects react to the message "play".

Objects and classes

- In analysis we should:
  - Discover the classes of objects involved (physical or abstract,) and
  - Identify the attributes of those classes.
- These translate into code as "class definitions"

```
public class ClassName
{
  Attribute definitions

  Method definitions
}
```

Example: Stereo

```
public class Stereo {
    // Attributes
    float volume;
    boolean radio_on;
    boolean cd_in;
    int current_song;

    // Methods
    void play_cd()
    {
        radio_on = false;
        if (cd_in) {
            current_song = 1;
        }
        // ...
    }
    void set_volume(float v)
    {
        volume = v;
    }
    // ...
}
```

Class definition structure

- Attribute definitions

  `type variable``;`

where *`type`* is either a primitive data type (`int`, `boolean`, etc.) or the name of a user-defined class.

Class definition structure (contd.)

- Method definitions

```
type method_name(list_of_parameters)
{
    statements;
}
```

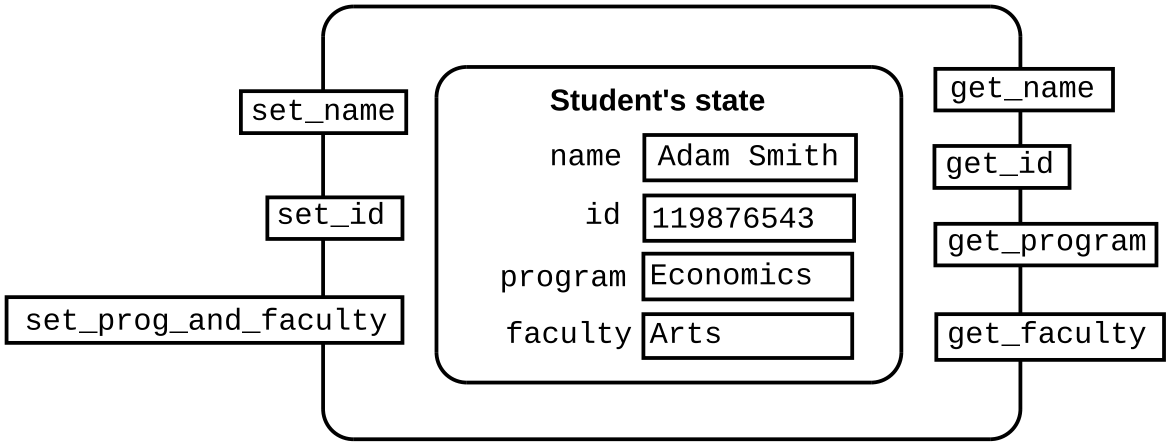
where *type* is either `void` (the method doesn't return anything,) a primitive data type or a user-defined data type. The *list_of_parameters* is of the form

```
type1 arg1, type2 arg2, ..., typen argn
```

Example

```
public class Student
{
    String name;
    long id;
    String program;
    String faculty;

    void set_name(String s)
    {
        name = s;
    }

    void set_id(long num)
    {
        id = num;
    }

    // Continues below ...
    String get_name()
    {
        return name;
    }

    long get_id()
    {
        return id;
    }

    void set_prog_and_faculty(String p,
                              String f)
    {
        program = p;
        faculty = f;
    }

    String get_program()
    { return program; }

    String get_faculty()
    { return faculty; }
} // Class Student ends here.
```

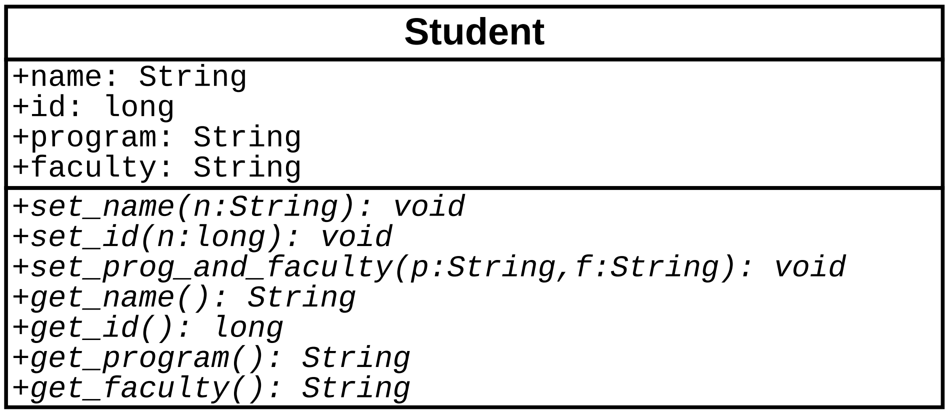
An object of the Student class



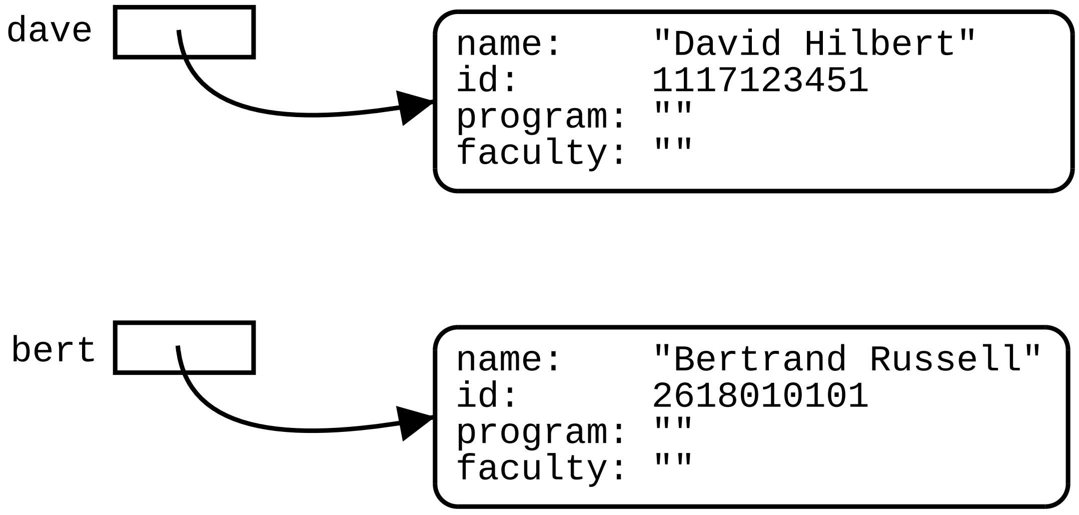
Objects are not classes

- A class can be thought of as a data type. Its values are objects.
- An *object* is an *instance* of a class.
- An object has its own separate identity and its own separate state.
- The *state* of an object is the values currently assigned to its attributes.
- Each object is stored in different memory locations.

Individual identity of objects



```
Student
+name: String
+id: long
+program: String
+faculty: String
+set_name(n:String): void
+set_id(n:long): void
+set_prog_and_faculty(p:String,f:String): void
+get_name(): String
+get_id(): long
+get_program(): String
+get_faculty(): String
```



Dealing with objects

- To be able to use a class and its objects we must be able to do three things:
  - Create instances of a class (i.e. new objects)
  - Access attributes of a given object (previously created)
  - Ask or tell a given object (previously created) to perform an operation (by sending a message to it, i.e. applying a method.)

Creating objects

- To create objects of a given class:

  First: Declare a variable of that type:

  ```
  class_name variable;
  ```

  Second: Assign the variable a new instance, using the new keyword:

  ```
  variable = new class_name();
  ```

- Example

  ```
  Student dave;

  dave = new Student();
  ```

- The two can be done in one line:

  ```
  Student bert = new Student();
  ```

Accessing attributes

- The attributes of an object can be accessed directly using the dot operator:

  `variable.attribute`

  ...but only if the attribute exists in the class of the variable.

- Example:

```
dave.name = “David Hilbert”;
dave.id = 1117123451;
System.out.println(dave.name);
System.out.println(dave.id);

bert.name = “Bertrand Russell”;
bert.id = 2618010101;
System.out.println(bert.name);
System.out.println(bert.id);
```

Sending messages to objects

- To interact with an object we send it a message by *calling*, or *invoking* one its methods.

- Calling a method is done by using the dot operator, and passing parameters or arguments (if any):

  ```
  variable.method_name(arguments)
  ```

  where the type of *`variable`* is a class which has a method called *`method_name`*, and *`arguments`* is a coma-separated list of values whose type matches those of the method's parameters.

Sending messages (contd.)

- For example:

  ```
  bert.set_prog_and_faculty(‘‘Philosophy’’, ‘‘Arts’’);

  dave.set_id(009876543);
  ```

- A method call

  ```
  a.m(b, c, d);
  ```

  could be interpreted as “sending the message `m` to the object `a` with arguments `b`, `c`, and `d`.”

Example

```
// in a file called Student.java
public class Student
{
    String name;
    long id;
    String program;
    String faculty;

    void set_name(String s)
    {
        name = s;
    }

    void set_id(long num)
    {
        id = num;
    }

    // Continues below ...
    String get_name()
    {
        return name;
    }

    long get_id()
    {
        return id;
    }

    void set_prog_and_faculty(String p,
                              String f)
    {
        program = p;
        faculty = f;
    }

    String get_program()
    { return program; }

    String get_faculty()
    { return faculty; }
} // Class Student ends here.
```

Example

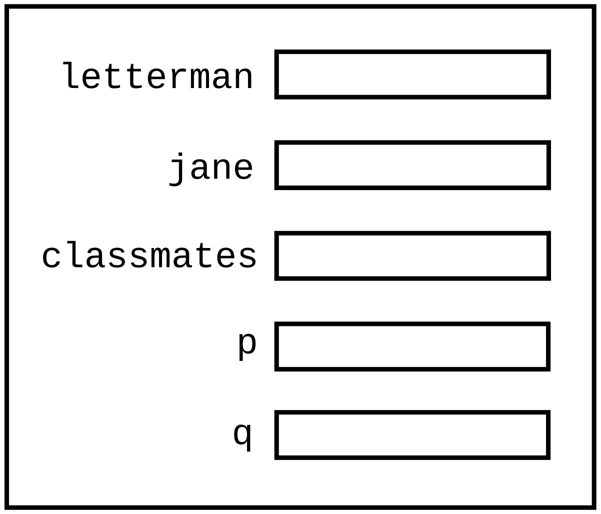
```
// in a file called StudentTester.java
import cs1.Keyboard;
public class StudentTester {
  public static void main(String[] args) {

    Student letterman, jane, p, q;
    boolean classmates;

    letterman = new Student();
    jane = new Student();       // Different studen

    letterman.set_name(``David'');
    letterman.set_id(000000011);

    jane.set_name(``Jane'');
    jane.set_id(9867554);

    letterman.set_prog_and_faculty(``Broadcasting'',
                                   ``Medicine'');
    jane.set_prog_and_faculty(``Physics'', ``Science'')

    p = letterman.get_program();
    q = jane.get_program();
```

```
      if (p.equals(q)) classmates = true; else classm
    }
  }
```

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();       // Different studen

letterman.set_name(‘‘David’’);
letterman.set_id(000000011);

jane.set_name(‘‘Jane’’);
jane.set_id(9867554);

letterman.set_prog_and_faculty(‘‘Broadcasting’’,
                               ‘‘Medicine’’);
jane.set_prog_and_faculty(‘‘Physics’’, ‘‘Science’’

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

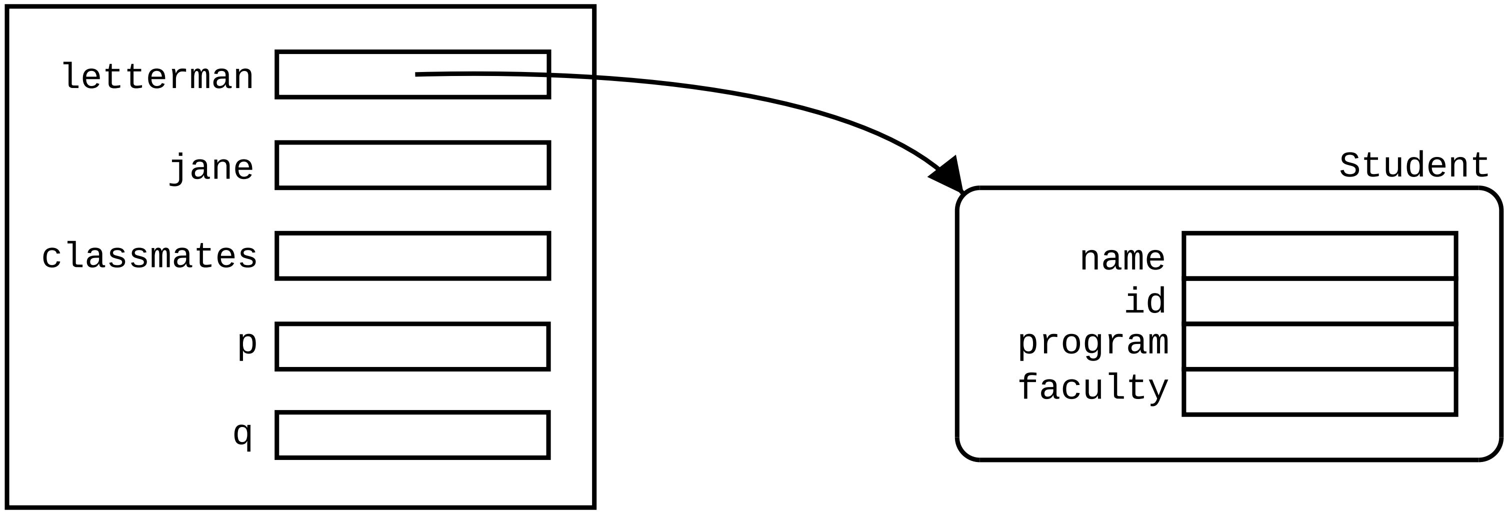
Example



Example

```
Student letterman, jane
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(‘‘David’’);
letterman.set_id(000000011);

jane.set_name(‘‘Jane’’);
jane.set_id(9867554);

letterman.set_prog_and_faculty(‘‘Broadcasting’’,
                                ‘‘Medicine’’);
jane.set_prog_and_faculty(‘‘Physics’’, ‘‘Science’’

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else classm
```

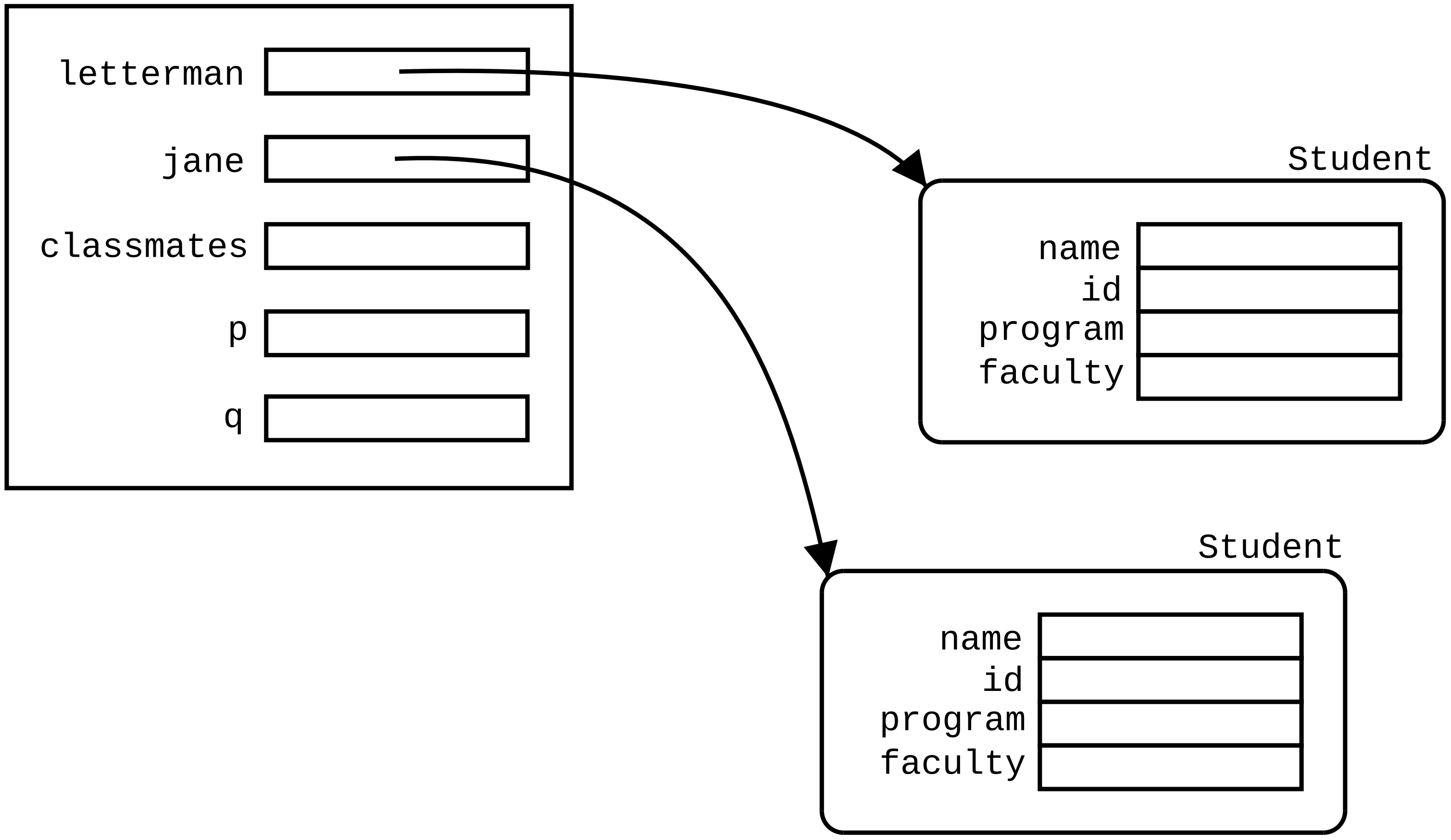
Example


letterman
jane
classmates
p
q
Student
name
id
program
faculty

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();       // Different studen

letterman.set_name(‘‘David’’);
letterman.set_id(000000011);

jane.set_name(‘‘Jane’’);
jane.set_id(9867554);

letterman.set_prog_and_faculty(‘‘Broadcasting’’,
                                ‘‘Medicine’’);
jane.set_prog_and_faculty(‘‘Physics’’, ‘‘Science’’

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

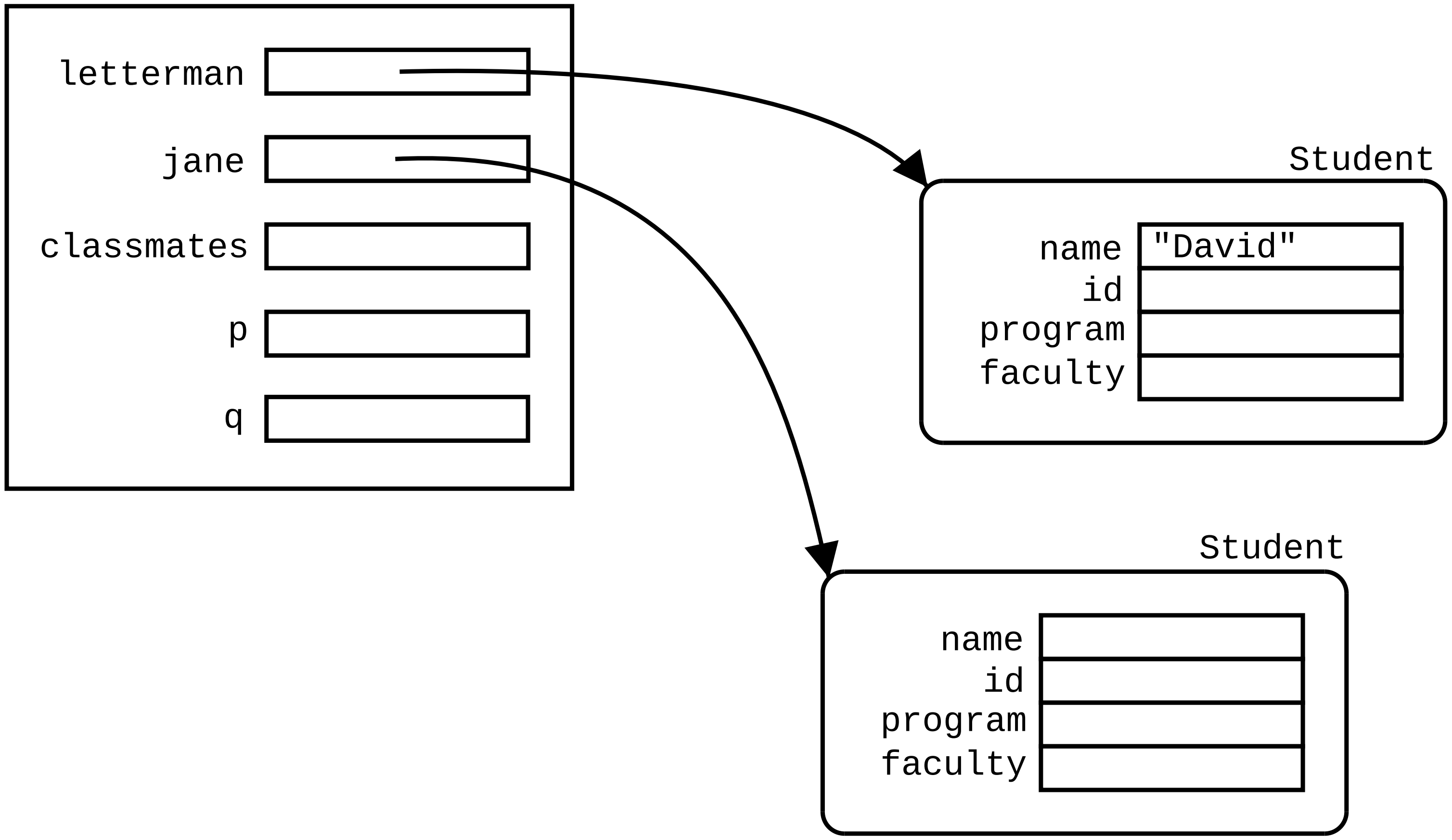
Example


letterman
jane
classmates
p
q
Student
name
id
program
faculty
Student
name
id
program
faculty

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(‘‘David’’);
letterman.set_id(000000011);

jane.set_name(‘‘Jane’’);
jane.set_id(9867554);

letterman.set_prog_and_faculty(‘‘Broadcasting’’,
                                ‘‘Medicine’’);
jane.set_prog_and_faculty(‘‘Physics’’, ‘‘Science’’

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

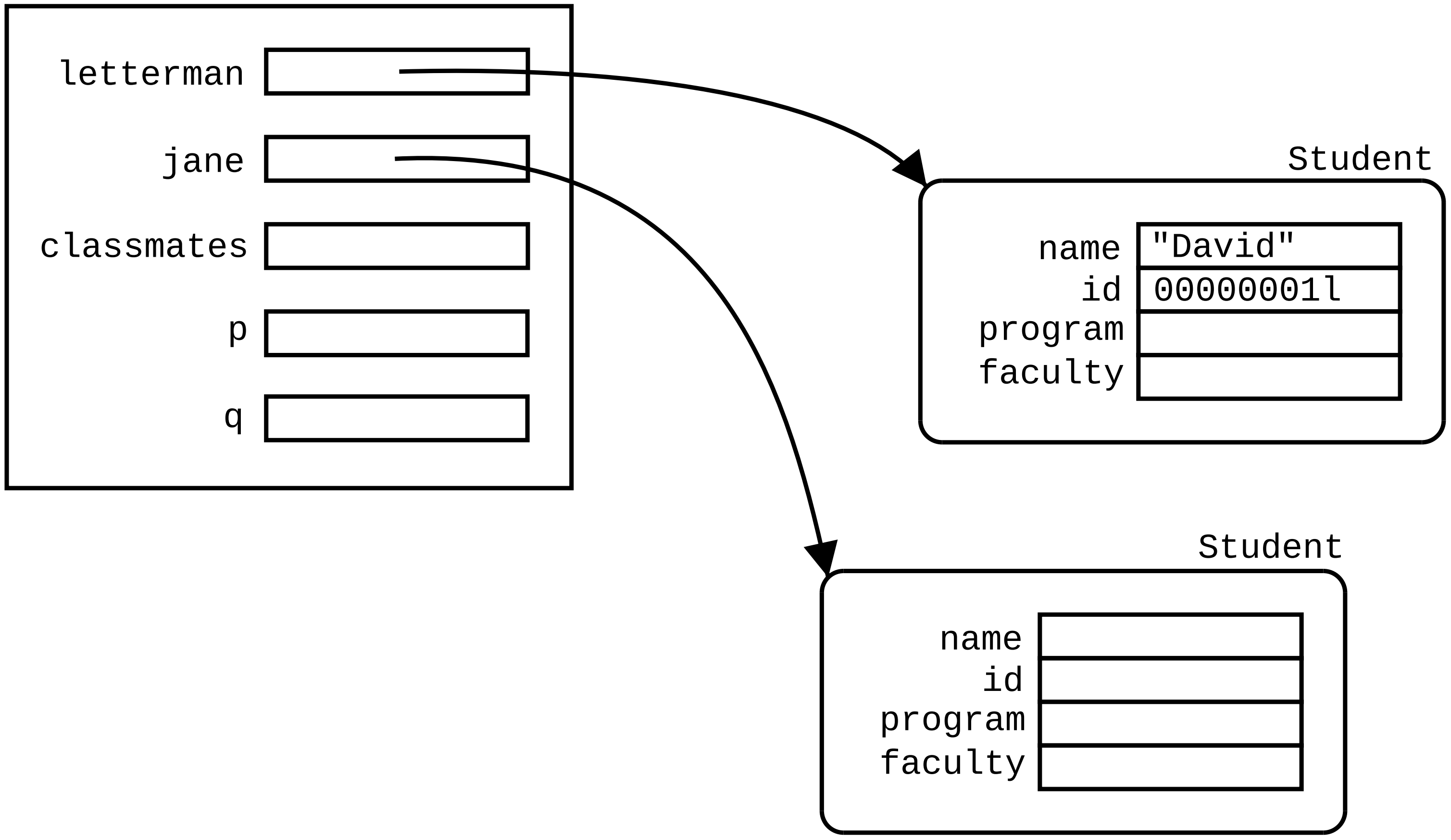
Example


letterman
jane
classmates
p
q
Student
name
"David"
id
program
faculty
Student
name
id
program
faculty

Example

```
Student letterman, jane;
String  p, q;
boolean classmates;

letterman = new Student();
jane = new Student();       // Different studen

letterman.set_name(``David'');
letterman.set_id(000000011);

jane.set_name(``Jane'');
jane.set_id(9867554);

letterman.set_prog_and_faculty(``Broadcasting'',
                               ``Medicine'');
jane.set_prog_and_faculty(``Physics'', ``Science'')

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

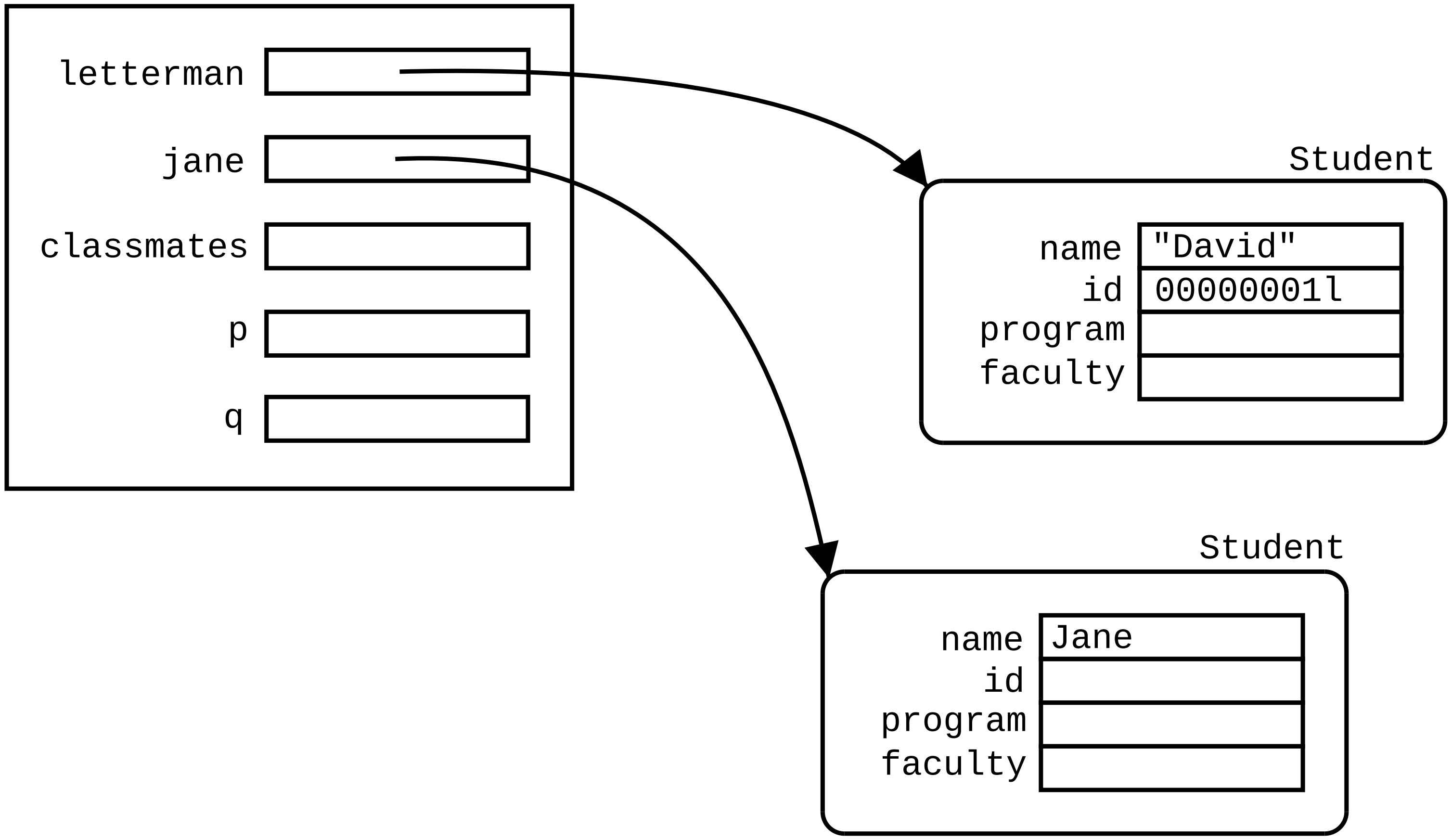
Example


letterman
jane
classmates
p
q
Student
name
id
program
faculty
"David"
00000001l
Student
name
id
program
faculty

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(``David'');
letterman.set_id(000000011);

jane.set_name(``Jane'');
jane.set_id(9867554);

letterman.set_prog_and_faculty(``Broadcasting'',
                               ``Medicine'');
jane.set_prog_and_faculty(``Physics'', ``Science'')

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

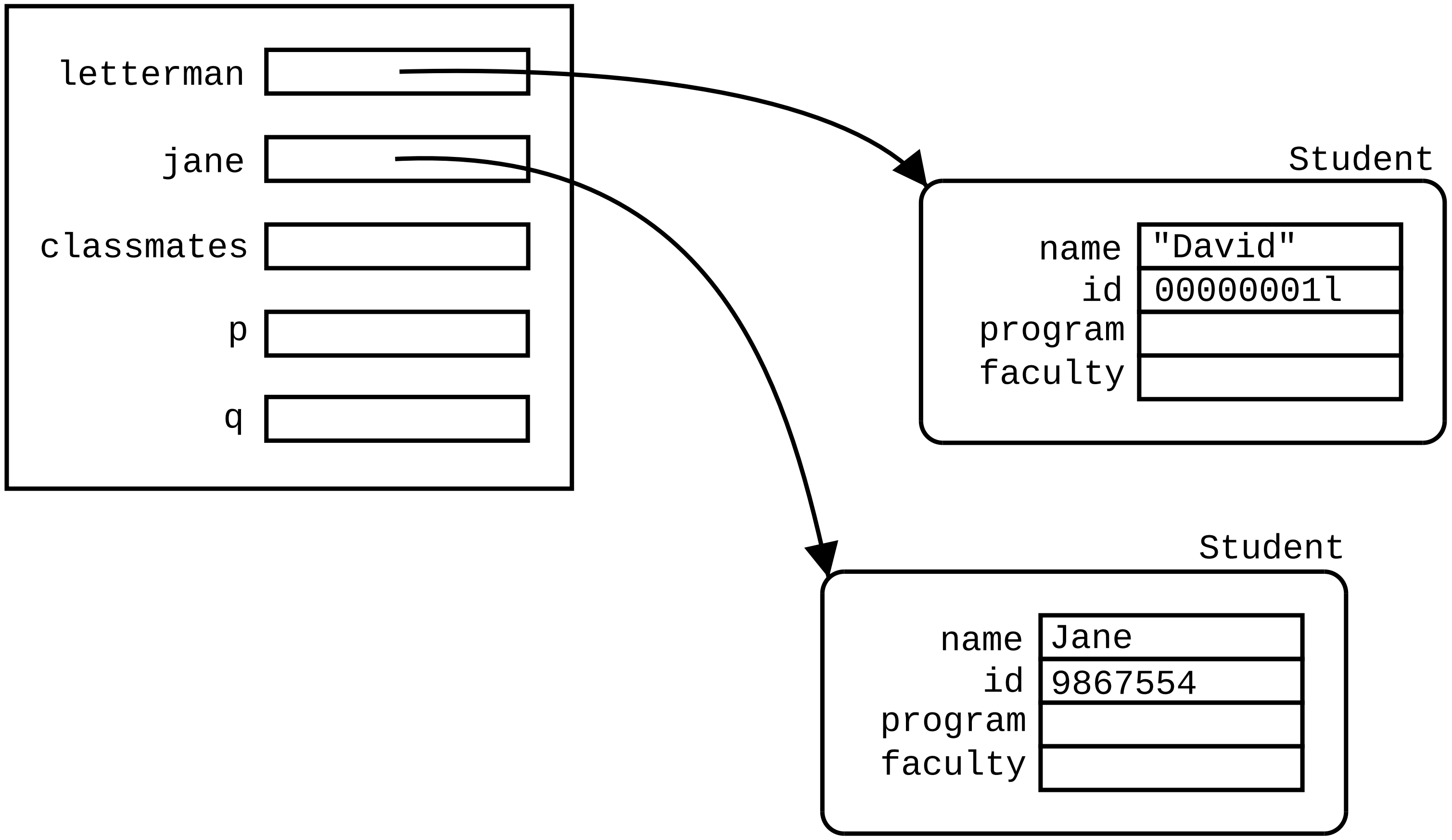
Example


letterman
jane
classmates
p
q
Student
name
"David"
id
00000001l
program
faculty
Student
name
Jane
id
program
faculty

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(``David'');
letterman.set_id(000000011);

jane.set_name(``Jane'');
jane.set_id(9867554);

letterman.set_prog_and_faculty(``Broadcasting'',
                               ``Medicine'');
jane.set_prog_and_faculty(``Physics'', ``Science'')

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

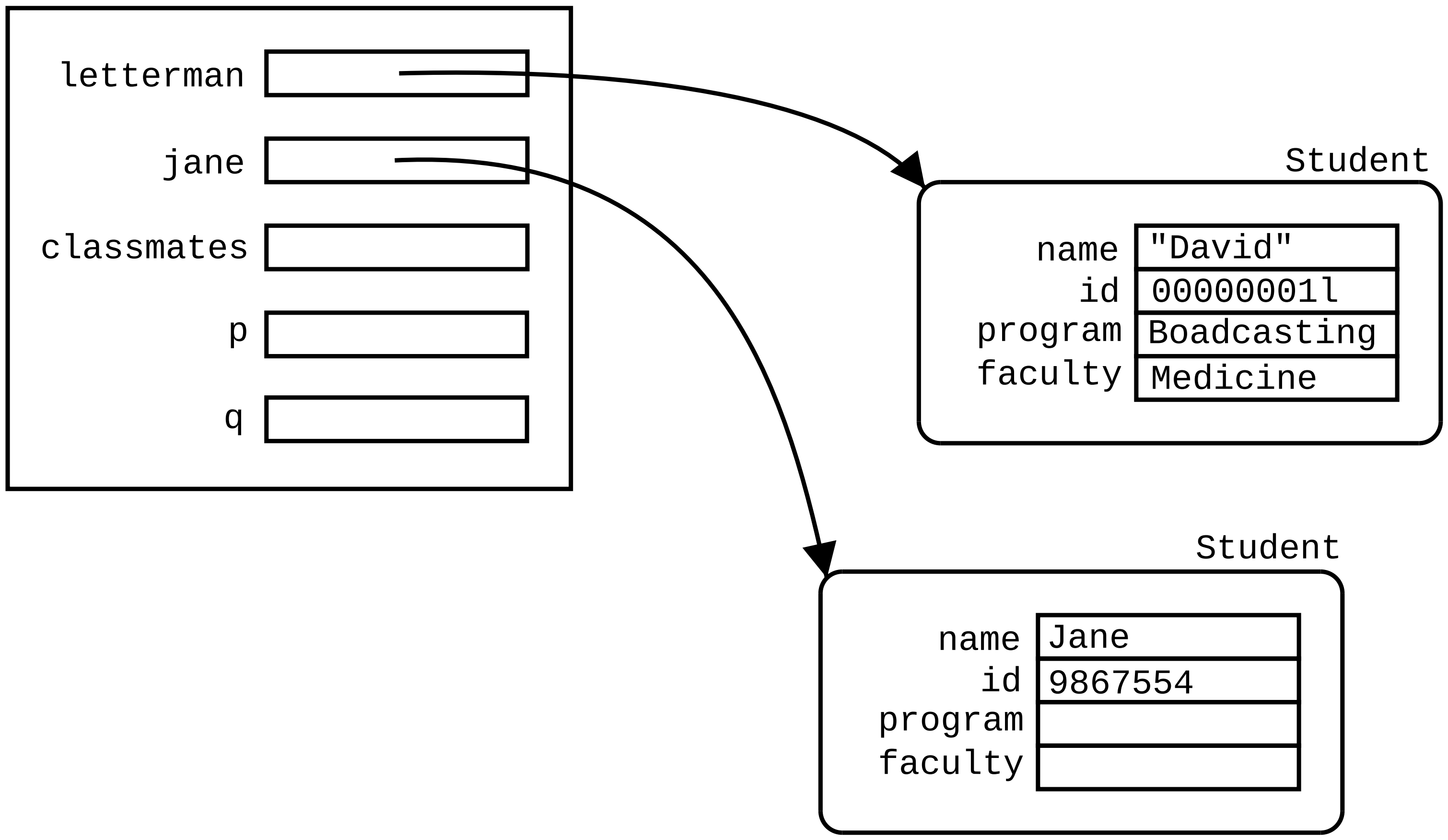
Example


letterman
jane
classmates
p
q
Student
name
"David"
id
00000001l
program
faculty
Student
name
Jane
id
9867554
program
faculty

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(‘‘David’’);
letterman.set_id(000000011);

jane.set_name(‘‘Jane’’);
jane.set_id(9867554);

letterman.set_prog_and_faculty(‘‘Broadcasting’’,
                               ‘‘Medicine’’);
jane.set_prog_and_faculty(‘‘Physics’’, ‘‘Science’’

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

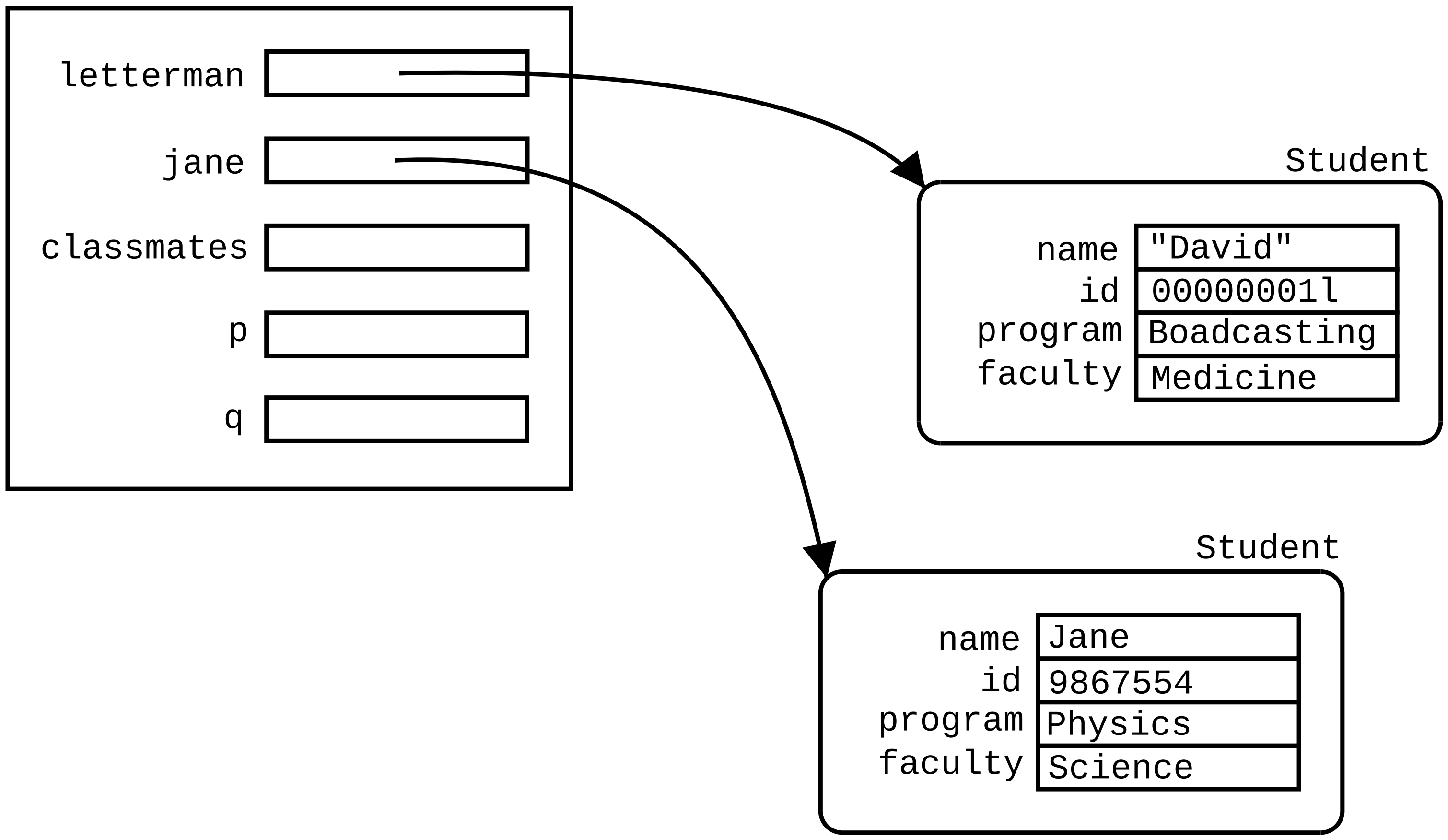
Example


letterman
jane
classmates
p
q
Student
name
id
program
faculty
"David"
00000001l
Boadcasting
Medicine
Student
name
id
program
faculty
Jane
9867554

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(``David'');
letterman.set_id(000000011);

jane.set_name(``Jane'');
jane.set_id(9867554);

letterman.set_prog_and_faculty(``Broadcasting'',
                               ``Medicine'');
jane.set_prog_and_faculty(``Physics'', ``Science''

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

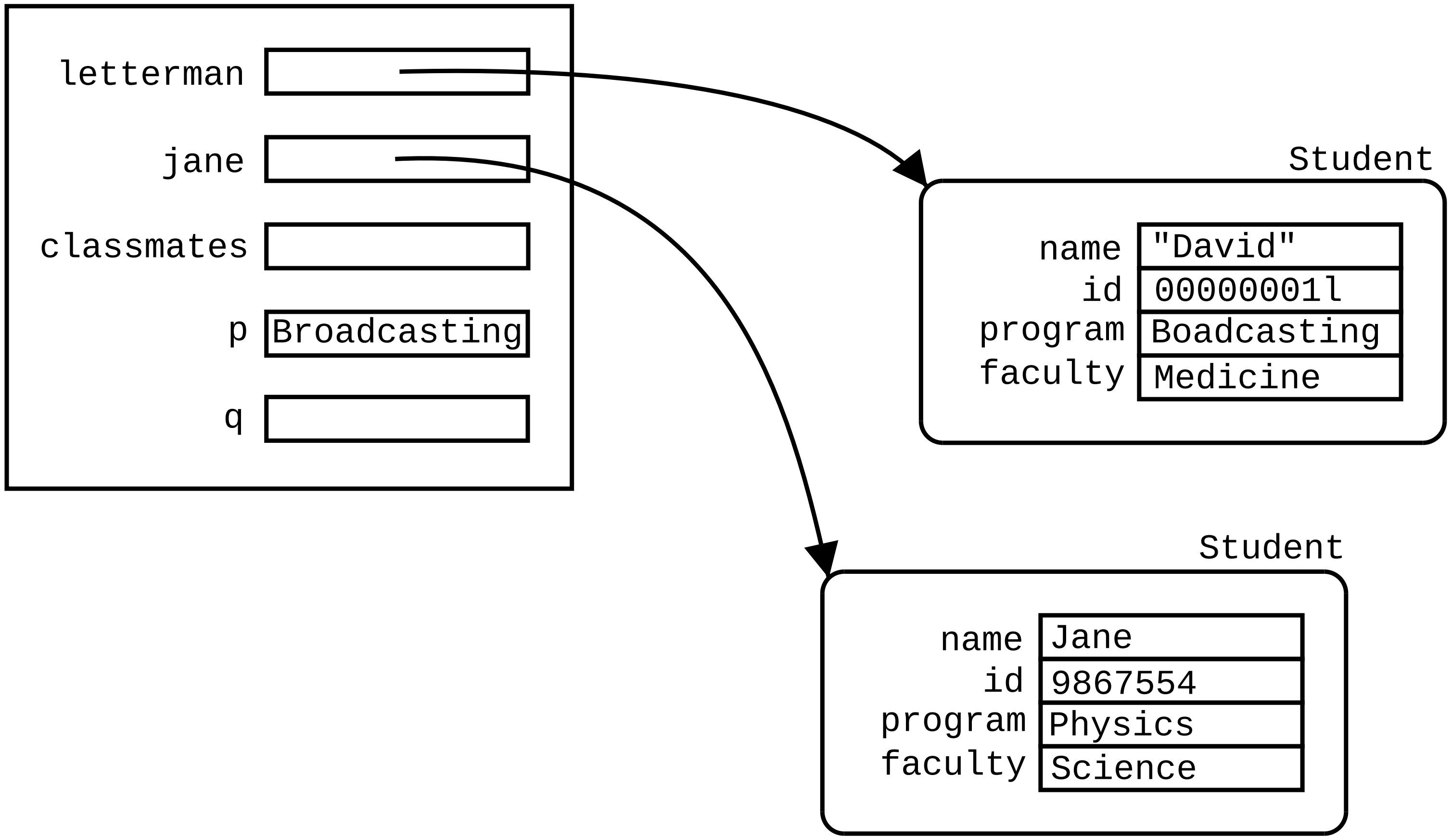
Example


letterman
jane
classmates
p
q
Student
name
id
program
faculty
"David"
00000001l
Boadcasting
Medicine
Student
name
id
program
faculty
Jane
9867554
Physics
Science

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(‘‘David’’);
letterman.set_id(000000011);

jane.set_name(‘‘Jane’’);
jane.set_id(9867554);

letterman.set_prog_and_faculty(‘‘Broadcasting’’,
                               ‘‘Medicine’’);
jane.set_prog_and_faculty(‘‘Physics’’, ‘‘Science’’

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

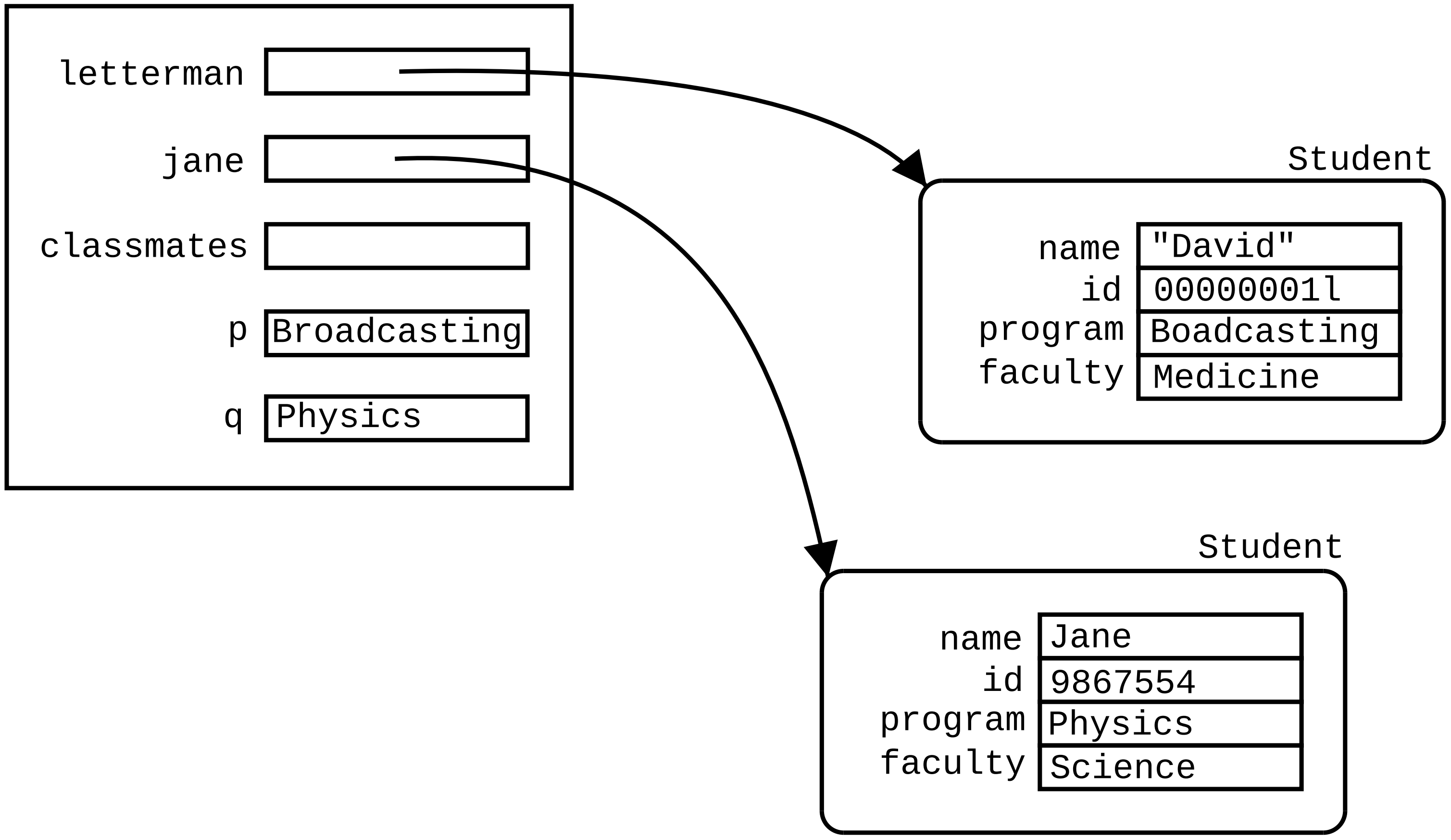
Example


letterman
jane
classmates
p
Broadcasting
q
Student
name
"David"
id
00000001l
program
Boadcasting
faculty
Medicine
Student
name
Jane
id
9867554
program
Physics
faculty
Science

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();      // Different studen

letterman.set_name(‘‘David’’);
letterman.set_id(000000011);

jane.set_name(‘‘Jane’’);
jane.set_id(9867554);

letterman.set_prog_and_faculty(‘‘Broadcasting’’,
                               ‘‘Medicine’’);
jane.set_prog_and_faculty(‘‘Physics’’, ‘‘Science’’

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

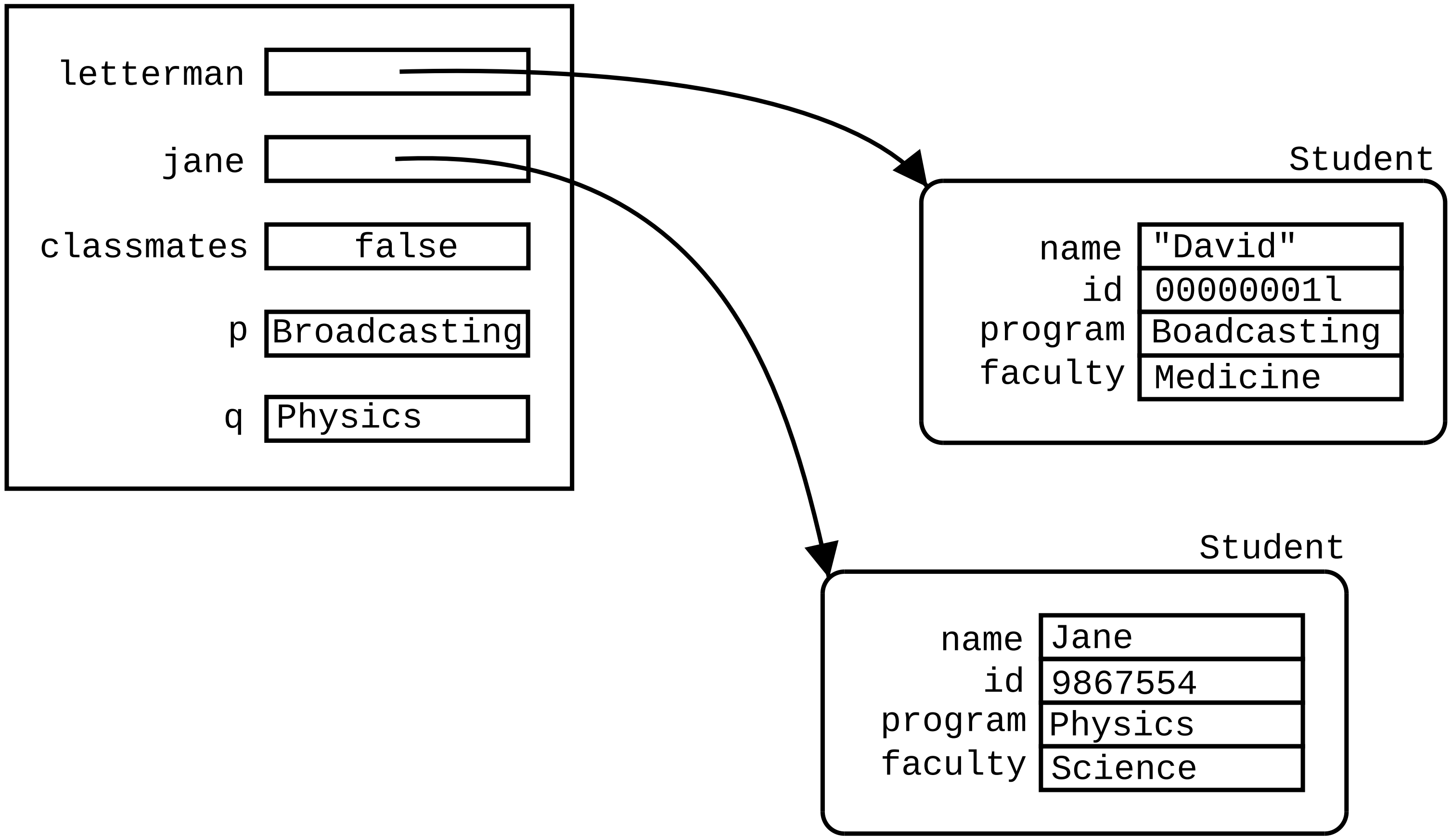
Example


letterman
jane
classmates
p
Broadcasting
q
Physics
Student
name
"David"
id
00000001l
program
Boadcasting
faculty
Medicine
Student
name
Jane
id
9867554
program
Physics
faculty
Science

Example

```
Student letterman, jane;
String p, q;
boolean classmates;

letterman = new Student();
jane = new Student();       // Different studen

letterman.set_name(“David”);
letterman.set_id(000000011);

jane.set_name(“Jane”);
jane.set_id(9867554);

letterman.set_prog_and_faculty(“Broadcasting”,
                               “Medicine”);
jane.set_prog_and_faculty(“Physics”, “Science”

p = letterman.get_program();
q = jane.get_program();

if (p.equals(q)) classmates = true; else class
```

Example



letterman
jane
classmates
false
p
Broadcasting
q
Physics
Student
name
"David"
id
00000001l
program
Boadcasting
faculty
Medicine
Student
name
Jane
id
9867554
program
Physics
faculty
Science

Method calls in context

- There are two forms of method calls:
  - Method call as a statement
  - Method call as an expression

- A method call is a statement if its return type is `void`, otherwise it is an expression.

- If a method call is an expression, it must appear in a context that allows expressions, such as:

  A. the right hand-side of an assignment:

  ```
  long n = dave.get_id();
  String s = dave.get_program();
  ```

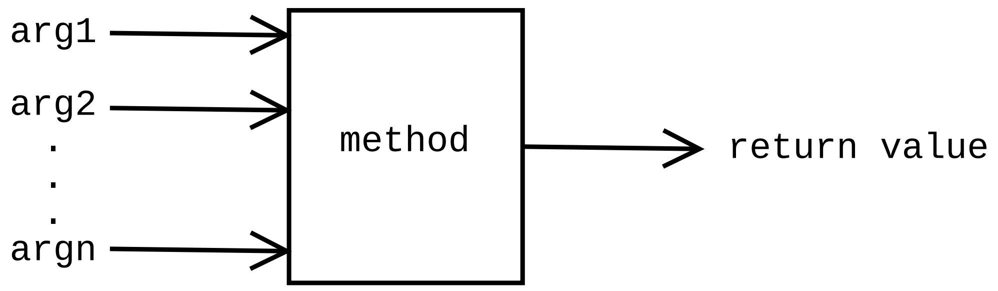
  B. ...or, the argument of another method:

  ```
  System.out.println(dave.get_id());
  bert.set_id(dave.get_id());
  ```

- But the types **must** match!

Methods as functions

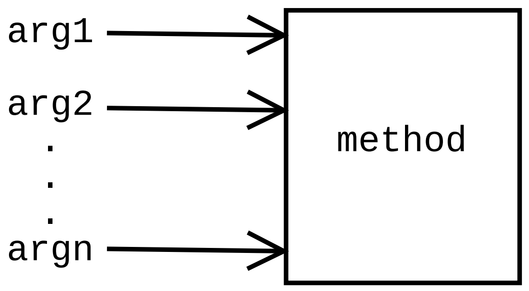
- Methods can be viewed as a "black box" with inputs and outputs:



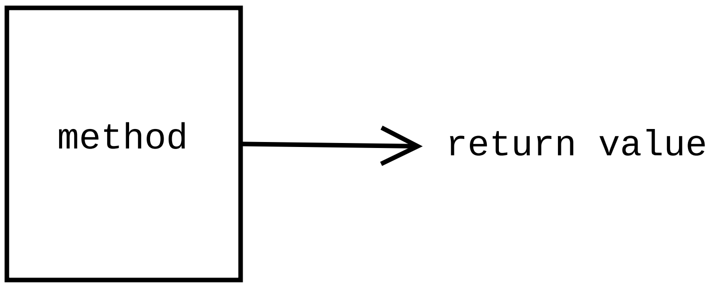
- There are three kinds of methods:
  - Mutators: Modify the state of objects,
  - Accessors: Return information about the object,
  - Constructors: Initialize a newly created object.

Method types

- Mutators are usually `void` methods, which do not return anything, but modify the state of the object:



- Accessor methods may only return values without expecting any arguments as input:



The end