



< HarvardLawyer

Main Page → Problems → Solve a Problem

MinMaxAccount >

Show Header

○ BJP4 Exercise 9.4: MonsterTruck

Language/Type:

Java classes implementing inheritance instance methods

Related Links: Car.java

Author: Marty Stepp (on 2016/09/08)

Suppose that the following two classes have been declared:

```
public class Car {
    public void m1() {
        System.out.println("car 1");
    public void m2() {
        System.out.println("car 2");
    }
    public String toString() {
        return "vroom";
    }
}
public class Truck extends Car {
    public void m1() {
        System.out.println("truck 1");
    }
    public void m2() {
        super.m1();
    public String toString() {
        return super.toString() + super.toString();
    }
}
```

Write a class MonsterTruck whose methods have the behavior below. Don't just print/return the output; whenever possible, use inheritance to reuse behavior from the superclass.

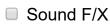
Method	Output/Return
m1	monster 1
m2	truck 1 car 1
toString	"monster vroomvroom"

```
Type your solution here:
 1 public class MonsterTruck extends Truck {
 2
       public void m1() {
           System.out.println("monster 1");
 3
 4
 5
       public void m2() {
           super.m1();
 6
           super.m2();
 7
 8
 9
10
       public String toString() {
11
           return "monster " + super.toString();
       }
12
13
14
15 }
```

This is an **inheritance problem.** Write a Java class using inheritance. (You do not need to write any import statements.)

Submit







Go to the next problem: MinMaxAccount

"monster vroomvroom"
"monster vroomvroom" console output:

problems/tests, etc.), please contact us.

result: pass

If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor. If something seems wrong with the site (errors, slow performance, incorrect

Is there a problem? Contact a site administrator.





< Janitor

Main Page → Problems → Solve a Problem

MonsterTruck >

Show Header

BJP4 Exercise 9.3: HarvardLawyer

Language/Type:

Java <u>classes implementing inheritance instance methods</u>

Related Links: <u>Lawyer.java</u>

Author: Marty Stepp (on 2016/09/08)

Write a class HarvardLawyer to accompany the other law firm classes described in this chapter. Harvard lawyers are like normal lawyers, but they make 20% more money than a normal lawyer, they get 3 days more vacation, and they have to fill out four of the lawyer's forms to go on vacation. That is, the getVacationForm method should return "pinkpinkpink". Make sure to interact with the superclass as appropriate.

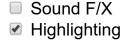
```
Type your solution here:
```

```
1 public class HarvardLawyer extends Lawyer {
2
       public double getSalary() {
 3
           return super.getSalary() * 1.2;
4
5
       public int getVacationDays() {
6
           return super.getVacationDays() + 3;
7
8
       public String getVacationForm() {
           return super.getVacationForm()+super.getVacationForm()
9
               +super.getVacationForm()+super.getVacationForm();
10
11
       }
12
13 }
```

This is an **inheritance problem.** Write a Java class using inheritance. (You do not need to write any import statements.)







Go to the next problem: MonsterTruck

test #1: getSalary 1

console output: 48000.0

result: **⊘** pass

test #2: getSalary 2 (change base salary to \$56000.00)

console output: 67200.0

result: **⊘** pass

test #3: getVacationDays 1

console output: 18

result: opass

test #4: getVacationDays 2 (change base days to 18)

console output: 26

result: opass

test #5: getVacationForm 1

console output: "pinkpinkpinkpink"

result: opass

test #6: getVacationForm 2 (change Lawyer form to green)

console output: "greengreengreengreen"

result: opass

test #7: sue

console output: I'll see you in court!

result: opass

If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor.

If something seems wrong with the site (errors, slow performance, incorrect problems/tests, etc.), please contact us.





< Marketer

<u>Main Page</u> → <u>Problems</u> → Solve a Problem

HarvardLawyer >

Show Header

O BJP4 Exercise 9.2: Janitor

Language/Type:

Java <u>classes implementing inheritance instance methods</u>

Related Links: <u>Employee.java</u>

Author: Marty Stepp (on 2016/09/08)

Write a class Janitor to accompany the other law firm classes described in this chapter. Janitors work twice as many hours per week as other employees (80 hours/week), they make \$30,000 (\$10,000 less than general employees), they get half as much vacation as other employees (only 5 days), and they have an additional method clean that prints "Workin' for the man." Make sure to interact with the superclass as appropriate.

```
Type your solution here:
```

```
1 public class Janitor extends Employee {
 2
       public double getSalary() {
 3
           return super.getSalary() - 10000.0;
 4
 5
       public int getHours() {
 6
           return super.getHours() * 2;
 7
 8
       public int getVacationDays() {
 9
           return super.getVacationDays() / 2;
10
11
       public void clean() {
           System.out.println("Workin' for the man.");
12
13
       }
14
15 }
```

This is an **inheritance problem.** Write a Java class using inheritance. (You do not need to write any import statements.)



Indent

■ Sound F/X



✓ Submit

You passed 7 of 7 tests.

Go to the next problem: HarvardLawyer

test #1: getSalary 1

console output: 30000.0

result: opass

test #2: getSalary 2 (change base salary to \$65535.00)

console output: 55535.0

result: opass

test #3: getHours 1

console output: 80

result: opass

test #4: getHours 2 (change base hours to 36)

console output: 72

result: **⊘** pass

test #5: getVacationDays 1

console output: 5

result: opass

test #6: getVacationDays 2 (change base days to 18)

console output: 9

result: opass

test #7: clean

console output: Workin' for the man.

result: opass

If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor.

If something seems wrong with the site (errors, slow performance, incorrect problems/tests, etc.), please contact us.

Is there a problem? Contact a site administrator.





< CarTruck2

<u>Main Page</u> → <u>Problems</u> → Solve a Problem

Janitor >

BJP4 Exercise 9.1: Marketer

Show Header

Language/Type:

Java classes implementing inheritance instance methods

Related Links: <u>Employee.java</u>

Author: Marty Stepp (on 2016/09/08)

Write the class Marketer to accompany the other law firm classes described in this chapter. Marketers make \$50,000 (\$10,000 more than general employees) and have an additional method called advertise that prints "Act now, while supplies last!" Make sure to interact with the Employee superclass as appropriate.

Type your solution here:

```
public class Marketer extends Employee {
   public double getSalary() {
      return super.getSalary() + 10000.0;
   }
   public void advertise() {
      System.out.println("Act now, while supplies last!");
   }
}
```

This is an **inheritance problem.** Write a Java class using inheritance. (You do not need to write any import statements.)









If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor.

If something seems wrong with the site (errors, slow performance, incorrect problems/tests, etc.), please contact us.

Is there a problem? Contact a site administrator.





< CarTruck

Main Page → Problems → Solve a Problem

Marketer >

O BJP4 Self-Check 9.9: CarTruck2

Language/Type:

Java <u>classes inheritance mystery</u>

Author:
Marty Stepp (on 2016/09/08)

Consider the following two classes:

```
public class Car {
    public void m1() {
        System.out.println("car 1");
    public void m2() {
        System.out.println("car 2");
    }
    public String toString() {
        return "vroom";
    }
}
public class Truck extends Car {
    public void m1() {
        System.out.println("truck 1");
    public void m2() {
        super.m1();
    }
    public String toString() {
        return super.toString() + super.toString();
    }
}
```

And assuming that the following variable has been declared:

```
Truck mytruck = new Truck();
```

What is the output from the following statements?

System.out.println(mytruck); vroomvroom

mytruck.m1(); truck 1

mytruck.m2(); car 1

Submit

Sound F/X

Go to the next problem: Marketer

#	question	your answer	result
1	<pre>System.out.println(mytruck);</pre>	vroomvroom	pass
2	<pre>mytruck.m1();</pre>	truck 1	pass
3	<pre>mytruck.m2();</pre>	car 1	pass

If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor.

If something seems wrong with the site (errors, slow performance, incorrect problems/tests, etc.), please contact us.

Is there a problem? Contact a site administrator.





< inheritanceVariableSyntax</p>
Main Page → Problems → Solve a Problem
CarTruck2 >

O BJP4 Self-Check 9.8: CarTruck

Language/Type:

Java <u>classes inheritance mystery</u>

Author:
Marty Stepp (on 2016/09/08)

Consider the following two classes:

```
public class Car {
    public void m1() {
        System.out.println("car 1");
    }

    public void m2() {
        System.out.println("car 2");
    }

    public String toString() {
        return "vroom";
    }
}

public class Truck extends Car {
    public void m1() {
        System.out.println("truck 1");
    }
}
```

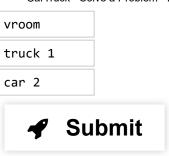
And assuming that the following variables have been declared:

```
Car mycar = new Car();
Truck mytruck = new Truck();
```

What is the output from the following statements?

Sound F/X

mytruck.m1();
mytruck.m2();



Go to the next problem: CarTruck2

#	question	your answer	result
1	<pre>System.out.println(mycar);</pre>	vroom	pass
2	mycar.m1();	car 1	pass
3	mycar.m2();	car 2	pass
4	<pre>System.out.println(mytruck);</pre>	vroom	pass
5	<pre>mytruck.m1();</pre>	truck 1	pass
6	<pre>mytruck.m2();</pre>	car 2	⊘ pass

If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor.

If something seems wrong with the site (errors, slow performance, incorrect problems/tests, etc.), please contact us.

Is there a problem? Contact a site administrator.





< subclassSyntax

Main Page → Problems → Solve a Problem

CarTruck >

BJP4 Self-Check 9.10: inheritanceVariableSyntax

Java classes inheritance syntax variables Language/Type:

Author: Marty Stepp (on 2016/09/08)

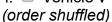
Consider the following classes:

```
public class Vehicle {...}
public class Car extends Vehicle {...}
public class SUV extends Car {...}
```

Which of the following are legal statements?

Sound F/X

- a. Car c = new Vehicle();
- b. ✓ Car c = new SUV();
- c. \square SUV s = new Car();
- d. ✓ Vehicle v = new Car();
- e. ✓ SUV s = new SUV();
- f. ✓ Vehicle v = new SUV();





You passed 1 of 1 tests.

Go to the next problem: CarTruck

```
Which of the following are legal statements?
      question #1:
                      Car c = new SUV();
      your answer:
Vehicle v = new Car();
SUV s = new SUV();
Vehicle v = \text{new SUV}();
             result:   opass
```

If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor.

If something seems wrong with the site (errors, slow performance, incorrect problems/tests, etc.), please contact us.

Is there a problem? Contact a site administrator.





Main Page → Problems → Solve a Problem

inheritanceVariableSyntax >

O BJP4 Self-Check 9.3: subclassSyntax

Which of the following is the correct syntax to indicate that class A is a subclass of B?

Sound F/X

```
a. public class B extends A {
b. public class A : super B {
c. public A implements B {
d. public A(super B) {
e. public class A extends B {
(order shuffled)
```



You passed 1 of 1 tests.

Go to the next problem: inheritanceVariableSyntax

question #1: Which of the following is the correct syntax to indicate that class A is

a subclass of B?

your answer: public class A extends B {

result: opass

If you do not understand how to solve a problem or why your solution doesn't work, please contact your TA or instructor.

If something seems wrong with the site (errors, slow performance, incorrect problems/tests, etc.), please contact us.

Is there a problem? Contact a site administrator.