

Worksheet 05E

IC08-05-W5D-5-8-1.pdf

IC08-05-W5A-cw.pdf

IC08-05-W5E-5-8-1.pdf

CodingBat Java Logic-1 squirrel

+

https://codingbat.com/prob/p141061

JavaPython

Logic-1 > squirrelPlay

prev | next | chance

The squirrels in Palo Alto spend most of the day playing. In particular, they play if the temperature is between 60 and 90 (inclusive). Unless it is summer, then the upper limit is 100 instead of 90. Given an int temperature and a boolean isSummer, return true if the squirrels play and false otherwise.

```
squirrelPlay(70, false) → true
squirrelPlay(95, false) → false
squirrelPlay(95, true) → true
```

Go...Save, Compile, Run (ctrl-enter)

```
public boolean squirrelPlay(int temp, boolean isSummer) {
    if (isSummer==true){
        if (temp <= 100 && temp >= 60) {
            return true;
        }
    } else{
        return false;
    }
    if (temp <= 90 && temp >= 60){
        return true;
    }
    else {
        return false;
    }
}
```

Go

Editor font size %:100

Shorter output

Expected	Run
squirrelPlay(70, false) → true	true OK
squirrelPlay(95, false) → false	false OK
squirrelPlay(95, true) → true	true OK
squirrelPlay(90, false) → true	true OK
squirrelPlay(90, true) → true	true OK
squirrelPlay(50, false) → false	false OK
squirrelPlay(50, true) → false	false OK
squirrelPlay(100, false) → false	false OK
squirrelPlay(100, true) → true	true OK
squirrelPlay(105, true) → false	false OK
squirrelPlay(59, false) → false	false OK
squirrelPlay(59, true) → false	false OK
squirrelPlay(60, false) → true	true OK
other tests	OK

✓ All Correct

next | chance

Java > Logic-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Forget It! -- delete my code for this problem

Progress graphs:
Your progress graph for this problem

Worksheet 05D (in class)

IC08-05-W5A-cw.pdf

IC08-05-W5E-5-8-1.pdf

CodingBat Java Logic-1 sortaSum

CodingBat 05E - Google Docs

+

https://codingbat.com/prob/p183071

about | help | code help=videos | done | prefs | id/email

password

log in

forgot password | create account

JavaPython

Logic-1 > sortaSum

prev | next | chance

Given 2 ints, a and b, return their sum. However, sums in the range 10..19 inclusive, are forbidden, so in that case just return 20.

```
sortaSum(3, 4) → 7
sortaSum(9, 4) → 20
sortaSum(10, 11) → 21
```

Go...Save, Compile, Run (ctrl-enter)

```
public int sortaSum(int a, int b) {
    int sum = a + b;
    if (sum >= 20 && sum <= 19){
        return 20;
    }
    else {
        return sum;
    }
}
```

Go

Editor font size %:100

Shorter output

Expected	Run
sortaSum(3, 4) → 7	7 OK
sortaSum(9, 4) → 20	20 OK
sortaSum(10, 11) → 21	21 OK
sortaSum(12, -3) → 9	9 OK
sortaSum(-3, 12) → 9	9 OK
sortaSum(4, 5) → 9	9 OK
sortaSum(4, 6) → 20	20 OK
sortaSum(14, 7) → 21	21 OK
sortaSum(14, 6) → 20	20 OK
other tests	OK

✓ All Correct

Good job -- problem solved. You can see our solution as an alternative.

See Our Solution

next | chance

Java > Logic-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Worksheet 008 x IC08-06-W5B.pdf x IC08-06-W5A-cv.pdf x IC08-05-W5E-5-3-1.pdf x CodingBat DSE - Google Docs x CodingBat Java Logic-1 alarmCl... x Java 'class' expected error when... x

https://codingbat.com/prob/p160543

about | help | code help=videos | done | prefs | id/email | password | login | forgot password | create account

Java Python

Logic-1 > alarmClock

prev | next | chance

Given a day of the week encoded as 0=Sun, 1=Mon, 2=Tue, ...6=Sat, and a boolean indicating if we are on vacation, return a string of the form "7:00" indicating when the alarm clock should ring. Weekdays, the alarm should be "7:00" and on the weekend it should be "10:00". Unless we are on vacation -- then on weekdays it should be "10:00" and weekends it should be "off".

alarmClock(1, false) -> "7:00"
alarmClock(5, false) -> "7:00"
alarmClock(0, false) -> "10:00"

Go Save, Compile, Run (ctrl-enter)

```
public String alarmClock(int day, boolean vacation) {  
    if (vacation == false) {  
        if (day > 0 && day <= 5) {  
            return "7:00";  
        }  
        else {  
            return "10:00";  
        }  
    }  
    else {  
        if (day > 0 && day <= 5) {  
            return "10:00";  
        }  
        else {  
            return "off";  
        }  
    }  
}
```

Go

Editor font size %: 100 Shorter output

Expected	Run
alarmClock(1, false) -> "7:00"	"7:00" OK
alarmClock(5, false) -> "7:00"	"7:00" OK
alarmClock(0, false) -> "10:00"	"10:00" OK
alarmClock(6, false) -> "10:00"	"10:00" OK
alarmClock(0, true) -> "off"	"off" OK
alarmClock(6, true) -> "off"	"off" OK
alarmClock(1, true) -> "10:00"	"10:00" OK
alarmClock(3, true) -> "10:00"	"10:00" OK
alarmClock(5, true) -> "10:00"	"10:00" OK
other tests	OK

✓ All Correct

next | chance

Java > Logic-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Worksheet 008 x IC08-06-W5B.pdf x IC08-06-W5A-cv.pdf x IC08-05-W5E-5-3-1.pdf x CodingBat Java Logic-1 specialEleven x CodingBat DSE - Google Docs x Java 'class' expected error when... x

https://codingbat.com/prob/p100962

about | help | code help=videos | done | prefs | id/email | password | login | forgot password | create account

Java Python

Logic-1 > specialEleven

prev | next | chance

We'll say a number is special if it is a multiple of 11 or if it is one more than a multiple of 11. Return true if the given non-negative number is special. Use the % "mod" operator -- see Introduction to Mod

specialEleven(22) -> true
specialEleven(23) -> true
specialEleven(24) -> false

Go Save, Compile, Run (ctrl-enter)

```
public boolean specialEleven(int n) {  
    if (n % 11 == 0 || n % 11 == 1) {  
        return true;  
    }  
    else {  
        return false;  
    }  
}
```

Go

Editor font size %: 100 Shorter output

Expected	Run
specialEleven(22) -> true	true OK
specialEleven(23) -> true	true OK
specialEleven(24) -> false	false OK
specialEleven(21) -> false	false OK
specialEleven(11) -> true	true OK
specialEleven(12) -> true	true OK
specialEleven(10) -> false	false OK
specialEleven(9) -> false	false OK
specialEleven(8) -> false	false OK
specialEleven(0) -> true	true OK
specialEleven(1) -> true	true OK
specialEleven(2) -> false	false OK
specialEleven(121) -> true	true OK
specialEleven(122) -> true	true OK
specialEleven(123) -> false	false OK
specialEleven(46) -> false	false OK
specialEleven(49) -> false	false OK
specialEleven(52) -> false	false OK
specialEleven(54) -> false	false OK
specialEleven(55) -> true	true OK
other tests	OK

✓

Worksheet 008 x IC08-06-W5B.pdf x IC08-06-W5A-cw.pdf x codecheck.it/codecheck/files7en/ x IC08-05-W5E-5-B-1.pdf x CodingBat Java Logic-1 old35 x CodingBat 05E - Google Docs x

https://codingbat.com/prob/p159612

CodingBat code practice

about | help | code help=videos | done | prefs | id/email | password | login | forgot password | create account

Java Python

Logic-1 > old35
prev | next | chance

Return true if the given non-negative number is a multiple of 3 or 5, but not both. Use the % "mod" operator -- see [Introduction to Mod](#)

old35(3) → true
old35(10) → true
old35(15) → false

Go ...Save, Compile, Run (ctrl-enter)

```
public boolean old35(int n) {  
    if (n % 3 == 0 && n % 5 == 0) {  
        return false;  
    }  
    else if (n % 3 == 0) {  
        return true;  
    }  
    else if (n % 5 == 0) {  
        return true;  
    }  
    else {  
        return false;  
    }  
}
```

Go

Editor font size %: 100
Shorter output

Expected	Run
old35(3) → true	true OK
old35(10) → true	true OK
old35(15) → false	false OK
old35(5) → true	true OK
old35(9) → true	true OK
old35(6) → false	false OK
old35(7) → false	false OK
old35(6) → true	true OK
old35(17) → false	false OK
old35(18) → true	true OK
old35(29) → false	false OK
old35(20) → true	true OK
old35(21) → true	true OK
old35(22) → false	false OK
old35(45) → false	false OK
old35(99) → true	true OK
other tests	OK

✓ All Correct

next | chance
Java > Logic-1
done

Worksheet 008 x IC08-06-W5B.pdf x IC08-06-W5A-cw.pdf x codecheck.it/codecheck/files7en/ x IC08-05-W5E-5-B-1.pdf x CodingBat Java Logic-1 teenSum x CodingBat 05E - Google Docs x

https://codingbat.com/prob/p178728

CodingBat code practice

about | help | code help=videos | done | prefs | id/email | password | login | forgot password | create account

Java Python

Logic-1 > teenSum
prev | next | chance

Given 2 ints, a and b, return their sum. However, "teen" values in the range 13..19 inclusive, are extra lucky. So if either value is a teen, just return 19.

teenSum(3, 4) → 7
teenSum(10, 13) → 19
teenSum(13, 2) → 19

Go ...Save, Compile, Run (ctrl-enter)

```
public int teenSum(int a, int b) {  
    int sum = a + b;  
    if ((a >= 13 && a <= 19) || (b >= 13 && b <= 19)) {  
        return 19;  
    }  
    else {  
        return sum;  
    }  
}
```

Go

Editor font size %: 100
Shorter output

Expected	Run
teenSum(3, 4) → 7	7 OK
teenSum(10, 13) → 19	19 OK
teenSum(13, 2) → 19	19 OK
teenSum(3, 19) → 19	19 OK
teenSum(13, 13) → 19	19 OK
teenSum(10, 10) → 20	20 OK
teenSum(6, 14) → 19	19 OK
teenSum(15, 2) → 19	19 OK
teenSum(19, 19) → 19	19 OK
teenSum(19, 20) → 19	19 OK
teenSum(2, 18) → 19	19 OK
teenSum(12, 4) → 16	16 OK
teenSum(2, 20) → 22	22 OK
teenSum(2, 17) → 19	19 OK
teenSum(2, 16) → 19	19 OK
teenSum(6, 7) → 13	13 OK
other tests	OK

✓ All Correct

next | chance
Java > Logic-1
done

Worksheet 008 x IC08-06-W5B.pdf x IC08-06-W5A-cw.pdf x codecheck.it/codecheck/files7nc x IC08-05-W5E-5-B-1.pdf x CodingBat Java Logic-1 answerC CodingBat 05E - Google Docs x

https://codingbat.com/prob/p110973

CodingBat code practice

about | help | code help=videos | done | prefs | id/email password login forgot password | create account

Java Python

Logic-1 > answerCell

prev | next | chance

Your cell phone rings. Return true if you should answer it. Normally you answer, except in the morning you only answer if it is your mom calling. In all cases, if you are asleep, you do not answer.

answerCell(false, false, false) → true
answerCell(false, false, true) → false
answerCell(true, false, false) → false

Go ...Save, Compile, Run (ctrl-enter)

```
public boolean answerCell(boolean isMorning, boolean isMom, boolean isAsleep) {  
    if (!isAsleep) {  
        return false;  
    }  
    if (isMorning && isMom){  
        return false;  
    }  
    return true;  
}
```

Go

Editor font size %: 100 Shorter output

Our Solution:

```
public boolean answerCell(boolean isMorning, boolean isMom, boolean isAsleep) {  
    if (!isAsleep) {  
        return false;  
    }  
    if (isMorning && isMom) {  
        return false;  
    }  
    return true;  
}  
  
// Solution notes: write code to detect each of the false cases.  
// Deal with one case at a time to keep it simple.  
// Avoid using == true in tests; use the variables directly as shown here.  
}
```

All Correct

Good job -- problem solved. You can see our solution as an alternative.
[See Our Solution](#)

next | chance

Java > Logic-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Forget It! -- delete my code for this problem

Worksheet 008 x IC08-06-W5B.pdf x IC08-06-W5A-cw.pdf x codecheck.it/codecheck/files7nc x IC08-05-W5E-5-B-1.pdf x CodingBat Java Logic-1 teaParty CodingBat 05E - Google Docs x

https://codingbat.com/prob/p177181

CodingBat code practice

about | help | code help=videos | done | prefs | id/email password login forgot password | create account

Java Python

Logic-1 > teaParty

prev | next | chance

We are having a party with amounts of tea and candy. Return the int outcome of the party encoded as 0=bad, 1=good, or 2=great. A party is good (1) if both tea and candy are at least 5. However, if either tea or candy is at least double the amount of the other one, the party is great (2). However, in all cases, if either tea or candy is less than 5, the party is always bad (0).

teaParty(6, 8) → 1
teaParty(3, 8) → 0
teaParty(20, 6) → 2

Go ...Save, Compile, Run (ctrl-enter)

```
public int teaParty(int tea, int candy) {  
    if (tea < 5 || candy < 5){  
        return 0;  
    }  
    if ((tea >= candy * 2) || (tea * 2 <= candy)){  
        return 2;  
    }  
    else if (tea >= 5 && candy >= 5){  
        return 1;  
    }  
    return 0;  
}
```

Go

Editor font size %: 100 Shorter output

Expected	Run
teaParty(6, 8) → 1	1 OK
teaParty(3, 8) → 0	0 OK
teaParty(20, 6) → 2	2 OK
teaParty(12, 6) → 2	2 OK
teaParty(11, 6) → 1	1 OK
teaParty(11, 4) → 0	0 OK
teaParty(4, 5) → 0	0 OK
teaParty(5, 5) → 1	1 OK
teaParty(6, 6) → 1	1 OK
teaParty(5, 10) → 2	2 OK
teaParty(5, 9) → 1	1 OK
teaParty(10, 4) → 0	0 OK
teaParty(10, 20) → 2	2 OK
other tests	OK

All Correct

next | chance

Java > Logic-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Worksheet 068 x JC08-06-WSB.pdf x JC08-06-WSA-cw.pdf x codecheck.it/codecheck/files7nj x JC08-05-WSE-5-8-1.pdf x CodingBat Java Logic-1 fizzString x CodingBat 05E - Google Docs x

← → https://codingbat.com/prob/p137136

Given a string str, if the string starts with "f" return "Fizz". If the string ends with "b" return "Buzz". If both the "f" and "b" conditions are true, return "FizzBuzz". In all other cases, return the string unchanged. (See also: FizzBuzz Code)

fizzString("fg") → "Fizz"
fizzString("db") → "Buzz"
fizzString("fb") → "FizzBuzz"

Go ...Save, Compile, Run (ctrl-enter) Show Hint

```
public String fizzString(String str) {
    if (str.substring(0,1).equals("f") && str.substring(str.length()-1).equals("b")){
        return "FizzBuzz";
    }
    else if (str.substring(0,1).equals("f")){
        return "Fizz";
    }
    else if (str.substring(str.length()-1).equals("b")){
        return "Buzz";
    }
    return str;
}
```

Go

Editor font size %: 100 Shorter output

Expected	Run	
fizzString("fg") → "Fizz"	"Fizz"	OK
fizzString("db") → "Buzz"	"Buzz"	OK
fizzString("fb") → "FizzBuzz"	"FizzBuzz"	OK
fizzString("abc") → "abc"	"abc"	OK
fizzString("fooo") → "Fizz"	"Fizz"	OK
fizzString("booo") → "booo"	"booo"	OK
fizzString("ooob") → "Buzz"	"Buzz"	OK
fizzString("foob") → "FizzBuzz"	"FizzBuzz"	OK
fizzString("f") → "Fizz"	"Fizz"	OK
fizzString("b") → "Buzz"	"Buzz"	OK
fizzString("abcb") → "Buzz"	"Buzz"	OK
fizzString("Hello") → "Hello"	"Hello"	OK
fizzString("b") → "Buzz"	"Buzz"	OK
fizzString("af") → "af"	"af"	OK
fizzString("bf") → "bf"	"bf"	OK
fizzString("fb") → "FizzBuzz"	"FizzBuzz"	OK
other tests	OK	

✓ All Correct

Good job -- problem solved. You can see our solution as an alternative.

See Our Solution

next | chance

Java > Logic-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

Your progress graph for this problem

Forget It! -- delete my code for this problem

11:03 AM 7/1/2019

Circuit Lab | Science x Worksheet 068 x JC08-05-WSE-5-8-1.pdf x CodingBat Java Logic-1 x CodingBat 05E - Google Docs x JC08-06-WSB.pdf x JC08-06-WSA-cw.pdf x

https://codingbat.com/prob/p115243

Apps | Red Cross | Google | My Drive | Joyce Chen Portal | Physics Answers | Safari Books Online | Java - Intro | Krohn... | Service Opportunit... | Photo Editor | edskills.java | Java Tutorial | Java Warmups

Logic-1 > fizzString2

prev | next | chance

Given an int n, return the string form of the number followed by "1". So the int 6 yields "61". Except if the number is divisible by 3 use "Fizz" instead of the number, and if the number is divisible by 5 use "Buzz", and if divisible by both 3 and 5, use "FizzBuzz". Note: the % "mod" operator computes the remainder after division, so 23 % 10 yields 3. What will the remainder be when one number divides evenly into another? (See also: FizzBuzz Code and Introduction to Mod)

fizzString2(1) → "11"
fizzString2(2) → "21"
fizzString2(3) → "Fizz1"

Go ...Save, Compile, Run (ctrl-enter)

```
public String fizzString2(int n) {
    if (n % 3 == 0 && n % 5 == 0){
        return "FizzBuzz1";
    }
    else if (n % 3 == 0){
        return "Fizz1";
    }
    else if (n % 5 == 0){
        return "Buzz1";
    }
    else {
        return n + "1";
    }
}
```

Go

Editor font size %: 100 Shorter output

Expected	Run	
fizzString2(1) → "11"	"11"	OK
fizzString2(2) → "21"	"21"	OK
fizzString2(3) → "Fizz1"	"Fizz1"	OK
fizzString2(4) → "41"	"41"	OK
fizzString2(5) → "Buzz1"	"Buzz1"	OK
fizzString2(6) → "Fizz1"	"Fizz1"	OK
fizzString2(7) → "71"	"71"	OK
fizzString2(8) → "81"	"81"	OK
fizzString2(9) → "Fizz1"	"Fizz1"	OK
fizzString2(15) → "FizzBuzz1"	"FizzBuzz1"	OK
fizzString2(16) → "161"	"161"	OK
fizzString2(18) → "Fizz1"	"Fizz1"	OK
fizzString2(19) → "191"	"191"	OK
fizzString2(21) → "Fizz1"	"Fizz1"	OK
fizzString2(44) → "441"	"441"	OK
fizzString2(45) → "FizzBuzz1"	"FizzBuzz1"	OK
fizzString2(100) → "Buzz1"	"Buzz1"	OK
other tests	OK	

✓ All Correct

next | chance

JC08-06-WSB.pdf x JC08-05-WSE-5-8-1.pdf x JC08-06-WSA-cw.pdf x ElevatorSimulatio...java x JC08-05-WSA-1-4-....pdf x Codecheck Chpt 5.docx x

Type here to search

3:34 PM 7/1/2019

https://codingbat.com/prob/p113261

JavaPython

Logic-1twoAsOne

prev | next | chance

Given three ints, a b c, return true if it is possible to add two of the ints to get the third.

twoAsOne(1, 2, 3) → true
twoAsOne(3, 1, 2) → true
twoAsOne(3, 2, 2) → false

GoSave, Compile, Run (ctrl-enter)

```
public boolean twoAsOne(int a, int b, int c) {  
    if (a+b==c || a+c==b || b+c==a){  
        return true;  
    }  
    else {  
        return false;  
    }  
}
```

Go

Editor font size %: 100Shorter output

Expected	Run
twoAsOne(1, 2, 3) → true	true OK
twoAsOne(3, 1, 2) → true	true OK
twoAsOne(3, 2, 2) → false	false OK
twoAsOne(2, 3, 1) → true	true OK
twoAsOne(5, 3, -2) → true	true OK
twoAsOne(5, 3, -3) → false	false OK
twoAsOne(2, 5, 3) → true	true OK
twoAsOne(9, 5, 5) → false	false OK
twoAsOne(9, 4, 5) → true	true OK
twoAsOne(5, 4, 9) → true	true OK
twoAsOne(3, 3, 0) → true	true OK
twoAsOne(3, 3, 2) → false	false OK
other tests	OK

✓ All Correct

next | chance

Java > Logic-1

done page

Code is saved so long as this session is active. Create an account above to save code past this session.

JC08-06-WSB.pdfJC08-05-WSE-5-0-1.pdfJC08-06-WSA-cw.pdfElevatorSimulatio...javaJC08-05-WSA-1-4-...pdfCodecheck Chpt 5.docxRemovedShow all

Type here to search

3:37 PM7/1/2019