Writing Methods Using Conditionals

Write the following 2 methods described below. Then, look at some solutions submitted by past students. Try to figure out whether those solutions would work. What **feedback** would you give them? Feel free to **edit your solution** if there's something that could be improved!

1. Given two int values, return their sum. Unless their sum is odd, then return their sum, times 1000, plus 1.

   oddOneOut(3, 10) → 13001
   oddOneOut(2, 10) → 12
   oddOneOut(21, 4) → 25001
   oddOneOut(13, 5) → 18

```
public static int oddOneOut(int a, int b)
{
    //Your code goes here
}
```

2. Given two int values, return their sum. Unless their sum is even, then return half the sum.

   gottaHalfSum(3, 10) → 13
   gottaHalfSum(2, 10) → 6
   gottaHalfSum(11, 4) → 15
   gottaHalfSum(13, 5) → 9

```
public static int gottaHalfSum(int a, int b)
{
    //Your code goes here
}
```

Feedback on Student Solutions

oddOneOut:

	Student Solution	Feedback you would give

A	`public static int OddOneOut(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if(sum % 2 == 1)`
	`  {`
	`  return sum * 1000 + 1;`
	`  }`
	`  else`
	`  {`
	`  return sum;`
	`  }`
	`}`	

B	`public static int oddOneOut(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if (sum % 2 == 1)`
	`  {`
	`  sum = sum * 1000 + 1;`
	`  }`
	`  return sum;`
	`}`	

C	`public static int oddOneOut(int a, int b)`
	`{`
	`  int oddOneOut = a+b;`
	`  if ((oddOneOut%2)!=0)`
	`  {`
	`  oddOneOut = ((a+b)*1000)+1;`
	`  }`
	`  return oddOneOut;`
	`}`	

D	`public static int oddOneOut(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if(sum % 2 != 0)`
	`  {`
	`  final = sum * 1000 + 1;`
	`  }`
	`  else`
	`  {`
	`  final = sum;`
	`  }`
	`  return final;`
	`}`	

E	```
	public static int oddOneOut(int a, int b)
	{
	  int sum = a + b;
	  int div = sum / 2;
	  if(sum != div * 2)
	  {
	     sum = sum * 1000 + 1;
	  }
	  return sum;
	}
	``	

gottaHalfSum:

	Student Solution	Feedback you would give

A	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  if (a + b % 2 == 0)`
	`  {`
	`  return (a + b)/2;`
	`  }`
	`  else`
	`  {`
	`  return a + b;`
	`  }`
	`}`	

B	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if (sum % 2 == 0)`
	`  {`
	`  return sum/2;`
	`  }`
	`  else`
	`  {`
	`  return sum;`
	`  }`
	`}`	

C	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if (sum % 2 = 0)`
	`  {`
	`  return sum/2;`
	`  }`
	`  else`
	`  {`
	`  return sum;`
	`  }`
	`}`	

D	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  if((a + b) % 2 == 0)`
	`  {`
	`  return (a + b) / 2;`
	`  }`
	`  if((a + b) % 2 != 0)`
	`  {`
	`  return a + b;`
	`  }`
	`}`	

E

```
public static int gottaHalfSum(int a, int b)
{
   int sum = a + b;
   if(sum % 2 == 0)
     {
        sum = sum / 2;
     }
   return sum;
}
```

F

```
public static int gottaHalfSum(int a, int b)
{
    int sum = a + b;
    if (sum % 2 == 0)
    {
          sum /= 2;
    }
    return sum;
}
```

G

```
public static int gottaHalfSum(int a, int b)
{
    num = a + b;
    if((num%2)==0)
        num=num/2;
        return num;
    else
        return num;
}
```

Feedback on Student Solutions

SOLUTIONS TO THE SOLUTIONS

oddOneOut:

	Student Solution	Feedback you would give

A	`public static int OddOneOut(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if(sum % 2 == 1)`
	`  {`
	`  return sum * 1000 + 1;`
	`  }`
	`  else`
	`  {`
	`  return sum;`
	`  }`
	`}`
		=)

B	`public static int oddOneOut(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if (sum % 2 == 1)`
	`  {`
	`  sum = sum * 1000 + 1;`
	`  }`
	`  return sum;`
	`}`
		=)

C	`public static int oddOneOut(int a, int b)`
	`{`
	`  int oddOneOut = a+b;`
	`  if ((oddOneOut%2)!=0)`
	`  {`
	`  oddOneOut = ((a+b)*1000)+1;`
	`  }`
	`  return oddOneOut;`
	`}`
		Don't name your variables the same as your methods. It will cause confusion down the road. (Technically, this one runs just fine.)

D	`public static int oddOneOut(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if(sum % 2 != 0)`
	`  {`
	`  final = sum * 1000 + 1;`
	`  }`
	`  else`
	`  {`
	`  final = sum;`
	`  }`
	`  return final;`
	`}`
		1) "final" is a reserved word in Java.... like public or int. It's the word Java uses for constants. Pick a different name for that variable.
		2) You need to declare and initialize the variable that will no longer be known as "final". Right now, the program wouldn't compile because that variable came out of nowhere. =)

E	```
	public static int oddOneOut(int a, int b)
	{
	  int sum = a + b;
	  int div = sum / 2;
	  if(sum != div * 2)
	  {
	     sum = sum * 1000 + 1;
	  }
	  return sum;
	}
	``
		=)

gottaHalfSum:

	Student Solution	Feedback you would give

A	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  if (a + b % 2 == 0)`
	`  {`
	`  return (a + b)/2;`
	`  }`
	`  else`
	`  {`
	`  return a + b;`
	`  }`
	`}`
		a + b % 2 == 0 isn't the same as (a+b) % 2 == 0... Think PEMDAS.

B	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if (sum % 2 == 0)`
	`  {`
	`  return sum/2;`
	`  }`
	`  else`
	`  {`
	`  return sum;`
	`  }`
	`}`
		=)

C	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  int sum = a + b;`
	`  if (sum % 2 = 0)`
	`  {`
	`  return sum/2;`
	`  }`
	`  else`
	`  {`
	`  return sum;`
	`  }`
	`}`
		= vs == error

D	`public static int gottaHalfSum(int a, int b)`
	`{`
	`  if((a + b) % 2 == 0)`
	`  {`
	`  return (a + b) / 2;`
	`  }`
	`  if((a + b) % 2 != 0)`
	`  {`
	`  return a + b;`
	`  }`
	`}`
		The method doesn't have a guaranteed return since all the returns are guarded by if conditions.

E

```
public static int gottaHalfSum(int a, int b)
{
   int sum = a + b;
   if(sum % 2 == 0)
     {
        sum = sum / 2;
     }
   return sum;
}
```

=)

F

```
public static int gottaHalfSum(int a, int b)
{
    int sum = a + b;
    if (sum % 2 == 0)
    {
          sum /= 2;
    }
    return sum;
}
```

=)

G

```
public static int gottaHalfSum(int a, int b)
{
    num = a + b;
    if((num%2)==0)
        num=num/2;
        return num;
    else
        return num;
}
```

1) Use { } if what follows the `if` or `else` is more than 1 line.

2) You didn't declare num, you just started using it. Java (unlike Python) won't let you do that. So the first line should have said: int num = ...

3) If you find the same line in both the if and else pieces, “factor it out” and put it just once outside the conditional.