

## 4 Instruction – Methods

The following exercises can be found on the website [codingbat.com](http://codingbat.com). Create an account if you had not done yet (so your work is kept over sessions).

1. Logic-2: blackjack <http://codingbat.com/prob/p117019>
2. Repeat the previous exercise, now with two extra methods as follows:

```
public int blackjack(int a, int b) {  
    return max(bj(a), bj(b));  
}  
  
int max(int a, int b) {  
  
}  
  
int bj(int n) {  
  
}
```

The method `blackjack` uses two other methods (functions), `max` and `bj`. Implement these methods as well (just type them below the `blackjack` method, as you see above, and fill in their bodies). Figure out for yourself what these functions should compute.

3. Logic-2: noTeenSum <http://codingbat.com/prob/p182879>
4. Logic-2: roundSum <http://codingbat.com/prob/p186753>
5. Logic-2: closeFar <http://codingbat.com/prob/p138990>  
Use the method `Math.abs` as mentioned in the description.
6. Array-3: canBalance <http://codingbat.com/prob/p158767>  
Include and use a method with header `int sum(int[] nums, int from, int upto)` that returns the sum of the elements of `nums` between `from` and `upto`, including `nums[from]` and not including `nums[upto]`. So the sum of the complete array is `sum(nums, 0, nums.length)`. An empty (sub)array should have sum 0. Note that the methods `canBalance` and `sum` can both have a parameter `nums` without problems.
7. \* Do the exercise above again, now with a more efficient implementation where the method `sum` is executed only once.