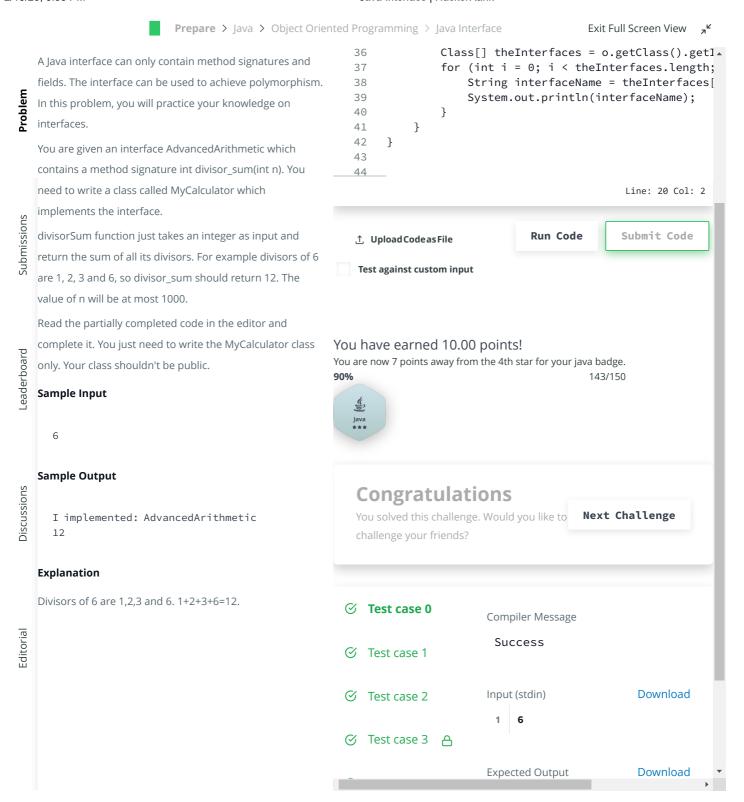
```
Prepare > Java > Object Oriented Programming > Java Method Overriding
                                                                                                         Exit Full Screen View
                                                              24
   When a subclass inherits from a superclass, it also
                                                              25
                                                                    }
   inherits its methods; however, it can also override the
                                                              26
                                                                    public class Solution{
                                                              27
   superclass methods (as well as declare and implement
                                                              28
   new ones). Consider the following Sports class:
                                                              29
                                                                         public static void main(String []args){
                                                              30
                                                                              Sports c1 = new Sports();
                                                                              Soccer c2 = new Soccer();
                                                              31
      class Sports{
                                                              32
                                                                              System.out.println(c1.getName());
           String getName(){
                                                                                                                Line: 23 Col: 2
               return "Generic Sports";
Submissions
           void getNumberOfTeamMembers(){
                                                                                                                Submit Code
                                                                                               Run Code
               System.out.println( "Each team has
                                                              }
                                                               Test against custom input
   Next, we create a Soccer class that inherits from the
   Sports class. We can override the getName method and
                                                          You have earned 10.00 points!
_eaderboard
   return a different, subclass-specific string:
                                                          You are now 97 points away from the gold level for your java badge.
                                                          3%
      class Soccer extends Sports{
           @Override
           String getName(){
               return "Soccer Class";
      }
                                                              Congratulations
   Note: When overriding a method, you should precede it
                                                              You solved this challenge. Would you like to
                                                                                                        Next Challenge
   with the @Override annotation. The parameter(s) and
                                                              challenge your friends?
   return type of an overridden method must be exactly
   the same as those of the method inherited from the
   supertype.

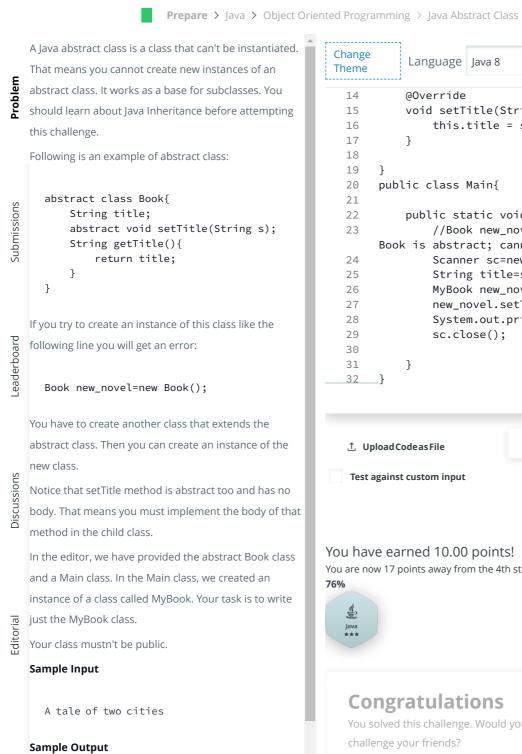
✓ Test case 0

                                                                                       Compiler Message
   Task
                                                                                        Success
   Complete the code in your editor by writing an
   overridden getNumberOfTeamMembers method that
   prints the same statement as the superclass'
                                                                                                                   Download
                                                                                       Expected Output
   getNumberOfTeamMembers method, except that it
                                                                                            Generic Sports
   replaces m{n} with m{11} (the number of players on a Soccer
                                                                                            Each team has n players in
   team).
                                                                                            Generic Sports
                                                                                            Saccar Class
   Output Format
```



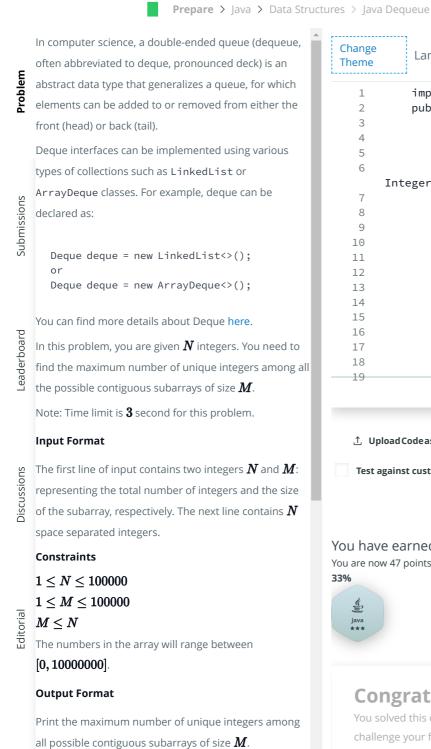
```
Prepare > Java > Data Structures > Java BitSet
    Java's BitSet class implements a vector of bit values (i.e.:
    false (0) or true (1)) that grows as needed, allowing us
    to easily manipulate bits while optimizing space (when
    compared to other collections). Any element having a
    bit value of \mathbf{1} is called a set bit.
    Given {f 2} BitSets, {m B_1} and {m B_2}, of size {m N} where all bits in
    both BitSets are initialized to oldsymbol{0}, perform a series of oldsymbol{M}
    operations. After each operation, print the number of
Submissions
    set bits in the respective BitSets as two space-separated
    integers on a new line.
    Input Format
    The first line contains {f 2} space-separated integers, {m N}
    (the length of both BitSets oldsymbol{B_1} and oldsymbol{B_2}) and oldsymbol{M} (the
   number of operations to perform), respectively.
    The m{M} subsequent lines each contain an operation in
    one of the following forms:
     • AND <set> <set>
     • OR <set> <set>
    • XOR <set> <set>
     FLIP <set> <index>
    • SET <set> <index>
    In the list above, \langle set \rangle is the integer 1 or 2, where 1
    denotes B_1 and 2 denotes B_2.
    <index> is an integer denoting a bit's index in the
    BitSet corresponding to <set>.
    For the binary operations AND, OR, and XOR,
   operands are read from left to right and the BitSet
    resulting from the operation replaces the contents of
    the first operand. For example:
       AND 2 1
    m{B_2} is the left operand, and m{B_1} is the right operand. This
    operation should assign the result of B_2 \wedge B_1 to B_2.
```

```
Exit Full Screen View
                  if(q.equals("OR")) {
19
                       b[left].or(b[right]);
20
21
     }
                  if(q.equals("XOR")) {
22
                       b[left].xor(b[right]);
23
24
     }
25
                  if(q.equals("FLIP")) {
                       b[left].flip(N - right - 1);
26
27
                  if(q.equals("SET")) {
28
                       b[left].set(N - right - 1);
29
30
                  System.out.println(b[0].cardinality()
31
     .cardinality());
32
              }
33
         }
34
     }
35
                                             Line: 35 Col: 4
                             Run Code
                                            Submit Code
Test against custom input
 Test case 0
                        Compiler Message
                         Success
 Test case 1
                                             Download
 Test case 2
                        Input (stdin)
 Test case 3
                             AND 1 2
                             SET 1 4
 Test case 4
                             FLIP 2 2
                             OR 2 1
 Test case 5
                        Expected Output
                                             Download
 Test case 6
                             0 0
```



```
Change
              Language Java 8
                                                     (0)
 Theme
   14
             @Override
   15
             void setTitle(String s){
   16
                  this.title = s;
   17
   18
   19
   20
        public class Main{
   21
             public static void main(String []args){
   22
   23
                  //Book new_novel=new Book(); This line pr
         Book is abstract; cannot be instantiated
                  Scanner sc=new Scanner(System.in);
   24
   25
                  String title=sc.nextLine();
   26
                  MyBook new_novel=new MyBook();
                  new_novel.setTitle(title);
   27
                  System.out.println("The title is: "+new_r
   28
   29
                  sc.close();
   30
   31
             }
   32
                                                 Line: 19 Col: 2
                                                Submit Code
                                 Run Code
   .↑. Upload Code as File
    Test against custom input
You have earned 10.00 points!
You are now 17 points away from the 4th star for your java badge.
76%
                                            133/150
   Congratulations
   You solved this challenge. Would you like to
                                          Next Challenge
   challenge your friends?
```

Exit Full Screen View



Sample Innut

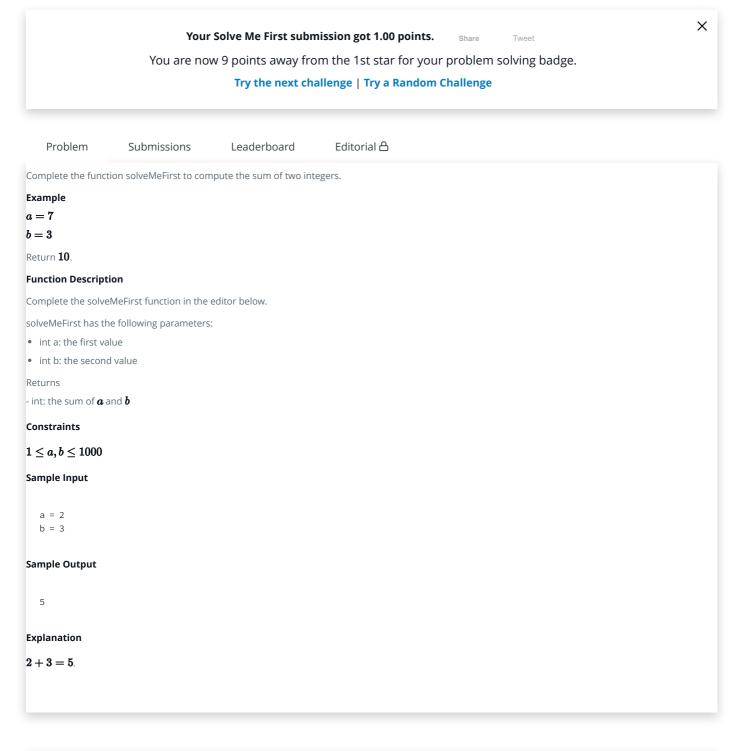
```
Change
             Language Java 8
                                                    (O)
 Theme
             import java.util.*;
    1
    2
             public class test {
                  public static void main(String[] args) {
    3
    4
                      Scanner in = new Scanner(System.in);
    5
                      final Deque<Integer> deque = new Arra
                        final Map<Integer, Integer> map = r
    6
         Integer>();
    7
                      final int n = in.nextInt();
                      final int m = in.nextInt();
    8
    9
                      int res =0;
   10
                      for (int i = 0; i < n; i++) {
   11
                           int num = in.nextInt();
   12
                             deque.addLast(num);
   13
   14
                      if (map.containsKey(num)) {
   15
                           map.put(num, map.get(num).intValu
   16
                      } else {
   17
                           map.put(num, 1);
   18
                      }
   19
                                                 Line: 8 Col: 19
                                                Submit Code
                                 Run Code
   .↑. Upload Code as File
    Test against custom input
You have earned 20.00 points!
You are now 47 points away from the 4th star for your java badge.
33%
                                           103/150
   Congratulations
   You solved this challenge. Would you like to
                                          Next Challenge
   challenge your friends?
```

Exit Full Screen View



## Solve Me First ★







```
static int solvementrst(int a, int b) {
TΩ
11
            // Hint: Type return a+b; below
12
               return a+b;
13
14
15
16
     public static void main(String[] args) {
            Scanner in = new Scanner(System.in);
17
             int _a = in.nextInt();
18
            int _b = in.nextInt();
19
            int sum = solveMeFirst(_a, _b);
20
21
            System.out.println(sum);
22
                                                                                                   Line: 12 Col: 11
                                                                                     Run Code
                                                                                                   Submit Code
Test against custom input
```

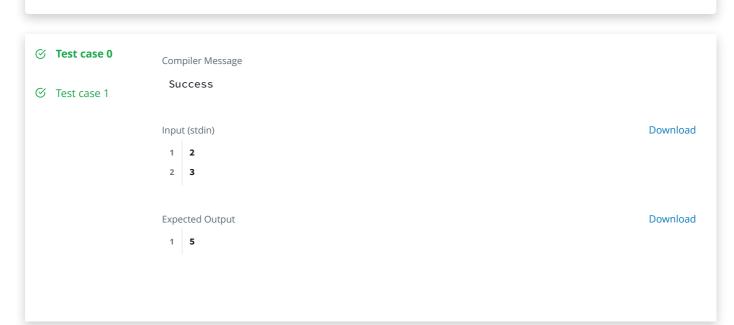
## You have earned 1.00 point!

You are now 9 points away from the 1st star for your problem solving badge. 70% 21/30





Next Challenge



Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy