# CS23333-Object Oriented Programming Using Java-2023



Quiz navigation

1 2 3

Show one page at a time

Finish review

| | |
|---|---|
| **Status** | Finished |
| **Started** | Tuesday, 8 October 2024, 3:36 PM |
| **Completed** | Tuesday, 8 October 2024, 3:39 PM |
| **Duration** | 3 mins 10 secs |

Question **1**

Correct

Marked out of 5.00

Flag question

Consider a sequence of the form 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149...

Write a method program which takes as parameter an integer n and prints the nth term of the above sequence. The nth term will fit in an integer value.

Example Input:

5

Output:

4

Example Input:

8

Output:

24

Example Input:

11

Output:

149

**For example:**

| Input | Result |
|---|---|
| 5 | 4 |
| 8 | 24 |
| 11 | 149 |

**Answer:** (penalty regime: 0 %)

```
import java.util.Scanner;

class fibo3{

    int a;
    int b;
    int c;
    fibo3(int a,int b,int c){
        this.a = a;
        this.b = b;
        this.c = c;
    }
    int nth(int x){
        if (x == 1){
            return 0;
        }
        else if(x == 2 && x == 3)
            return 1;
        else{
            int temp1,temp2,temp;
            int count = 4;
            while(x >= count){
                temp = this.a+this.b+this.c;
                temp1 = this.c;
                this.c = temp;
                temp2 = this.b;
                this.b = temp1;
                this.a = temp2;
                count++;
            }
            return this.c;
        }
    }
}
public class Main{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        int t = s.nextInt();
        fibo3 r = new fibo3(0,1,1);
        System.out.print(r.nth(t));
    }
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| | 5 | 4 | 4 | |
| | 8 | 24 | 24 | |
| | 11 | 149 | 149 | |

Passed all tests!

Question **2**

Correct

Marked out of 5.00

Flag question

You have recently seen a motivational sports movie and want to start exercising regularly. Your coach tells you that it is important to get up early in the morning to exercise. She sets up a schedule for you:

On weekdays (Monday - Friday), you have to get up at 5:00. On weekends (Saturday & Sunday), you can wake up at 6:00. However, if you are on vacation, then you can get up at 7:00 on weekdays and 9:00 on weekends.

Write a program to print the time you should get up.

Input Format

Input containing an integer and a boolean value.

The integer tells you the day it is (1-Sunday, 2-Monday, 3-Tuesday, 4-Wednesday, 5-Thursday, 6-Friday, 7-Saturday). The boolean is true if you are on vacation and false if you're not on vacation.

You have to print the time you should get up.

Example Input:

1 false

Output:

6:00

Example Input:

5 false

Output:

5:00

Example Input:

1 true

Output:

9:00

**For example:**

| Input | Result |
|---|---|
| 1 false | 6:00 |
| 5 false | 5:00 |
| 1 true | 9:00 |

**Answer:** (penalty regime: 0 %)

```
import java.util.Scanner;
  public class Main{
      public static void main(String[] args){
          Scanner sc =new Scanner(System.in);
          int day=sc.nextInt();
          boolean v=sc.nextBoolean();
          String t;
          if(v){
              if(day>1&&day<7)
              t="7:00";
              else
              t="9:00";
          }
          else{
              if(day>1&&day<7)
              t="5:00";
              else
              t="6:00";
          }
          System.out.println(t);
      }

  }
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| | 1 false | 6:00 | 6:00 | |
| | 5 false | 5:00 | 5:00 | |
| | 1 true | 9:00 | 9:00 | |

Passed all tests!

Question **3**

Correct

Marked out of 5.00

Flag question

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

4th term: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

And so on. Write a program that takes as parameter an integer n and prints the nth terms of this sequence.

Example Input:

1

Output:

1

Example Input:

4

Output:

1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

**For example:**

| Input | Result |
|---|---|

| Input | Result |
|---|---|
| 1 | 1 |
| 2 | 1 2 1 |
| 3 | 1 2 1 3 1 2 1 |
| 4 | 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1 |

**Answer:** (penalty regime: 0 %)

```java
import java.util.Scanner;

public class SequenceGenerator{

    public static void main(String[]args){
        Scanner S = new Scanner(System.in);
        int n = S.nextInt();
        String term = generateTerm(n);
        System.out.print(term);
    }
    private static String generateTerm(int n){
        if (n==1){
            return "1";
        }
        String prevTerm = generateTerm (n-1);
        StringBuilder currentTerm = new StringBuilder(prevTerm);
        currentTerm.append(" " + n + " ");
        currentTerm.append(prevTerm);
        return currentTerm.toString();
    }
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| | 1 | 1 | 1 | |
| | 2 | 1 2 1 | 1 2 1 | |
| | 3 | 1 2 1 3 1 2 1 | 1 2 1 3 1 2 1 | |
| | 4 | 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1 | 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1 | |

Passed all tests!

Finish review

◄ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►