

MethodPractice

Directions

Copy the following code into a file named MethodPractice1.java

```
import java.util.*;  
  
public class MethodPractice1  
{  
    private Scanner keyboard = new Scanner(System.in);  
    public static void main(String[] args)  
    {  
  
        System.out.println("*****");  
        System.out.println("    Test Method1");  
        System.out.println("*****");  
  
        System.out.println("Weight is " + method1(150));  
        System.out.println("Weight is " + method1(99));  
        System.out.println("Weight is " + method1(200));  
        System.out.println("Weight is " + method1(300));  
  
        System.out.println("\n\n*****");  
        System.out.println("    Test Method2");  
        System.out.println("*****");  
        app.method2("Computers are fun!");  
  
        System.out.println("\n\n*****");  
        System.out.println("    Test Method3");  
        System.out.println("*****");  
        System.out.println("\nCount = " + method3());  
  
        System.out.println("\n\n*****");  
        System.out.println("    Test Method4");  
        System.out.println("*****");  
        System.out.println("String = " + method4("red"));  
        System.out.println("String = " + method4("green"));  
        System.out.println("String = " + method4("blue"));  
  
        System.out.println("\n\n*****");  
        System.out.println("    Test Method5");  
        System.out.println("*****");  
        System.out.println("Random Number = " + method5(5));  
        System.out.println("Random Number = " + method5(50));
```

```
        System.out.println("Random Number = " + method5(500));
        System.out.println();
    }

/** This method returns a string using the following
 * conditions: if weight is less than 100 it returns
 * "small", if weight is greater than or equal to 100
 * and less than or equal to 200 it returns "medium",
 * if weight is greater than 200 it returns "large".
 * @return the string "small", "medium", or "large"
 * @param weight number representing a weight
 */
public String method1(int weight)
{

}

/** This method prints phrase 10 times.
 * @ param phrase the string to be printed
 */
public void method2(String phrase)
{

}

/** This method allows a user to enter an unknown
 * number of integers from the keyboard. When the
 * sentinel value -1 is entered the method returns
 * the count of the number of integers entered.
 * @return count of the number of integers entered
 */
public int method3()
{

}

/** This method returns a string containing the
 * first and last letter of str concatenated
 * together.
 * @return a string containing two letters
 * @param str the string from which to extract
 *          the first and last letters
 */
public String method4(String str)
```

```

{
}

/** This method returns a random number.
 *  @return a random number in range of 0 to upper-1
 *  @param upper the upper limit of the random number
 */
public int method5(int upper)
{
}

}

```

1. Implement **method1**. This method returns a string using the following conditions: if weight is less than 100 it returns "small", if weight is greater than or equal to 100 and less than or equal to 200 it returns "medium", and if weight is greater than 200 it returns "large".

The table below shows the return value of 4 calls made to method1.

Call	Return Value
method1(50)	small
method1(180)	medium
method1(220)	large
method1(100)	medium

2. Implement **method2**. This method prints the parameter phrase 10 times down the screen. For example, the call `method2("Our true mentor in life is science.")` would produce the following output:

```
Our true mentor in life is science.  
Our true mentor in life is science.
```

- Use a for loop in your implementation.
3. Implement **method3**. This method allows a user to enter an unknown number of integers from the keyboard. When the sentinel value -1 is entered the method displays the count of the number of integers entered. For example, if the user inputs 5 integers, not including the sentinel value, the method would return 5.
- Use a while loop in your implementation.
4. Implement **method4**. This method returns a string containing the first and last letter of str concatenated(merged) together.

The table below shows the return value of 4 calls made to **method4**.

Call	Return Value
<code>method4("dog")</code>	dg
<code>method4("cat")</code>	ct
<code>method4("horse")</code>	he
<code>method4("chicken")</code>	cn

5. Implement method5. This method returns a random number in the range of 0 to upper - 1. The table below shows the range of possible return values for 4 calls made to method5.

Call	Return Value Range
method5(2)	0-1
method5(10)	0-9
method5(150)	0-149
method5(200)	0-199