

26

Demonstrate the working of junit to reverse a word & using assert statement for proof of the value

Aim :- To understand the working of junit assert statement by comparing the reversed value with expected one.

```
import static org.junit.Assert.assertEquals;
import java.util.Scanner;
class SreeethaTest {
    public static void main (String [] args) {
        String str;
        char ch;
        Scanner sc = new Scanner (System.in);
        System.out.print ("Enter a string : ");
        str = sc.nextLine ();
        System.out.println ("Reverse of a
string " + str + " is : ");
        for (int i = str.length (); i > 0; --i)
            {
                System.out.print (str.charAt (i - 1));
                assertEquals ("mani", str);
                assertEquals ("mani", str);
            }
    }
}
```

Output :-

27

input .

mani

Actual output .

inam.

Test cases :-

Test case no : 1

Test case name : Expected one same as actual one.

Input = mani	Expected output	Actual output	Remarks
	inam.	inam.	SUCCESS.

Test case no : 2

Test case Name : Expected one same as actual one.

Input = Aman	Expected output	Actual output	Remarks
	rama.	r	Failure.

Exp:12

write a white box testing code (junit) to string
comparision of word and using assert statements
for proof the value.

Aim :- To understand the working of junit
assert statements by comparing two strings.

```
import static org.junit.Assert.assertEquals;  
import java.util.Scanner;  
public class third {  
    public static void main (String [] args)  
    {  
        Scanner in = new Scanner (System.in);  
        System.out.println ("enter the  
        username");  
        String str1 = in.nextLine();  
        System.out.println ("Reenter the username");  
        String str2 = in.nextLine();  
        assertEquals (str1, str2);  
    }  
}
```

Ex: 13 :
Write a junit code for voting system and
uses assert statement and verify the whole
box testing :

Aim :- To understand the working of 'and
true statements by checking the voting age

```
import static org.junit.Assert.assertEquals;
```

```
import java.util.Scanner;
```

```
class Four
```

```
{  
    public static void main (String [] args)
```

```
{  
    int age , shirt ;
```

```
    Scanner Scan = new Scanner (System . in );  
    System . out . println (" please enter your age ");  
    age = Scan . nextInt ();  
    if (age > = 18 )
```

```
{  
    System . out . println (" welcome to  
    voting system you can vote ");
```

```
System.out.println ("welcome  
else
```

```
{
```

```
shirt = (18 - age);
```

```
System.out.println ("sorry, you can  
vote after: " + shirt + " years");
```

```
assert True (age == shirt);
```

```
}
```

```
}
```

```
}
```

Exp: 14 :-

write a program using function to calculate the Simple Interest . Suppose the customer is a Senior citizen . He is being offering 12 percent rate of interest ; for all other customers , the ROI is 10 percent . The output values should verify using white box testing .

Aim :- write a program that calculates the Simple Interest based on percentage rate conditions and verify the result using assert true code .

```
import static org.junit.Assert.assertTrue;
```

```
import java.util.Scanner
```

```
class Interest {
```

```
}
```

```
public static void main (String [] args)
```

```
{
```

```
Scanner sc = new Scanner (System.in);
```

```
float p = sc.nextFloat();
```

```
float R = sc.nextFloat();
```

```
float r = sc.nextInt();
```

```
float si = (P*T*R)/100;
```

```
System.out.println("Simple Interest = ");
```

```
assert true (3600 == si);
```

```
}
```

```
{ public static void main
```

```
(String args) { System.out.println
```

```
"Hello World"; } }
```

```
class
```

```
new java.awt.Window().setVisible(true);
```

```
new JButton("Hello World");
```

```
frame.add(button);
```

```
(new JFrame("Hello World")) .show(); } }
```

checks whether the given number is palindrome or not and verify the output value should verify using white box testing.

Aim :- To check whether the given number is palindrome or not and verify the result using assert true code.

```
import java.util.Scanner;
import static org.junit.Assert.assertEquals;
public class Palindrome {
    public static void main (String args[])
    {
        Scanner in = new Scanner (System.in);
        int r, sum = 0, temp; int n = in.nextInt();
        int () ;
        temp = n ;
        while (n > 0)
        {
            r = n % 10; n = n / 10;
            sum = sum + r;
```

```
        sum = (sum * 10) + r ;
    }
    System.out.println(sum);
    assert True (787 == sum);

    if (temp == sum)
        System.out.println(sum + " is
palindrome number");
    else
        System.out.println(sum + " is not
palindrome number");
```

Ques :-

Write a program to convert decimal number equivalent to binary Number and octal

Numbers ? The output values should verify using white box testing ?

Aim :- To convert the decimal number to its equivalent binary number and octal number and the output values verified using Assert code .

```
import static org.junit.Assert.assertTrue;
```

```
import java.util.Scanner;
```

```
class binary
```

```
{
```

```
public static void main (String [] args)
```

```
{
```

```
Scanner in = new Scanner (System.in);
```

```
// decimal number
```

```
int decimal = in.nextInt();
```

```
// convert decimal to binary
```

string binary = integer . toBinaryString
(decimal);

System.out.println ("BINARY IS " + binary);

System.out.print ("OCTALS IS ");

System.out.println (integer . toOctal
String (decimal));

assert True (m = = decimal);

}

}