

Exp no 11

demonsrate the working of JUnit to reverse a word and using assert statement for proof of the value

Aim: To understand the working of JUnit assert statements by comparing the reversed value with expected one

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

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

```
class Saucedha test
```

```
{
```

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

```
{
```

```
    String str;
```

```
    char ch;
```

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

```
    System.out.println("Reverse of a string " + str + " is:");
```

```
    for (int j = str.length() - 1; j >= 0; --j)
```

```
{
```

```
        System.out.print(str.charAt(j) + " ");
```

```
    assertEquals("mani", str);
```

```
}
```

```
    assertEquals("mani", str);
```

```
}
```

```
}
```

Output

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
	Pran	Pran	Success

Test case no: 2

Test case name: Expected one same as actual one

Input = Amca	Expected output	Actual output	Remarks
	Pran	Pran	Failure

EXP NO: 12

Write a white box testing code (JUnit) to string comparison of word and using assert statement for proof the value.

AIM: To understand the working of JUnit assert statements by comparing two strings

```
import static org.junit.Assert.*;
import java.util.Scanner;
```

```
public class Word {
```

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

```
{
```

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

```
System.out.println("Enter the user name");
```

```
String str1 = in.nextLine();
```

```
System.out.println("Re-enter the user name");
```

```
String str2 = in.nextLine();
```

```
assert Equals(str1, str2);
```

```
}
```

Output

Input = Ame Expected output = Ame Actual output = Ame
Remarks = Successful

test case no 2

Input = Ame Expected output = Ameltra Actual output = Ameltra
Remarks = Unsuccessful

EXP NO: 13

write a junit code for voting system and use assert statement and verify the white box testing.

AIM: To understand the working of JUnit Test statements by checking the voting age.

import static org.junit.Assert.*; import java.util.Scanner;

class JUnit {

{
 public static void main (String[] args)

{
 int age; 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");

 }

 else

 {

 int short = (18 - age);

 System.out.println ("Sorry - you can vote after: " + short + " years");

 assert true (age == short);

333

output

Test case 1

Input = 19 Expected output = you can vote Actual output = you can vote

Remarks : successful

Test case 2

Input = 15 Expected output : you can vote after 3 years

Actual output = you can vote after 3 years

Remarks : successful

Exp NO: 14

AIM: write a program that calculates the simple interest based on the percentage rate conditions and verify the result

```
import static org.junit.Assert.*;
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 T = sc.nextFloat();
```

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

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

```
assert True (3600 == SI);
```

```
}
```

3

Output

Test case

Input: 600 Expected output: 3600.0 Actual output: 3600.0
600
1

Remarks: Successful

Test case 2:

Input: 600 Expected output: 1080.0 Actual output: 1080.0
60
B

Remarks: Successful

ERP NO IS

ATM: To check whether the given numbers
palindrome or not and verify the result
using assertTrue code.

import java.util.Scanner;

import static org.junit.Assert.assertTrue;

public class palindrome

{

public static void main (String args[])

{

Scanner in = new Scanner (System.in);

int d, sum = 0, temp; int n = in.nextInt();

temp = n;

while (n > 0)

{

d = n % 10; n = n / 10;

sum = (sum * 10) + d;

}

System.out.println (sum);

assertTrue (787 == sum)

PF (temp == sum)


```

        System.out.println (sum+" is palindrome number");
    else
        System.out.println (sum+" is not palindrome number");
}
}

```

Output:

Test case 1

Input: 787 Expected output: 787 is a palindrome

Expected output: 787 is a palindrome.

Remarks: Successful.

Test case 2

Input = 143 Expected output: 143 is not a palindrome

Actual output = 341

Remarks Unsuccessful

EXP NO 16

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.True;

```

```

class Binary

```

```

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

```

```

        Scanner in = new Scanner (System.in);

```

```

        int decimal = in.nextInt();

```

```

        String binary = Integer.toBinaryString(decimal);

```

```

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

```

```

        System.out.println (Integer.toOctalString (decimal));

```

```

        assertTrue (14 == decimal);
    }
}

```

Output

Test case 1

Input: 14 Expected output: 140 Binary is 1110
Octal is 16

Actual output: Binary 1110
Octal is 16. Remark: Success

Test case 2

Input: 15 Expected output: Binary is ~~1001~~ 1111
Octal is 17

Actual output: Binary is 1111
Octal is 17

Remarks: Success