

Java Programs

WEDNESDAY

APRIL

13

WK 16 • 104 262

All programs inside
psvm(s...)

1) Max of 3 Numbers

if ($a > b$ && $a > c$)

sop("a is big No.");

elseif ($b > a$ && $b > c$)

sop("b is big No.");

else

sop("c is big No.");

2) Swap two no. without using 3rd variable.

$n_1 = n_1 + n_2;$

$n_2 = n_1 - n_2;$

$n_1 = n_1 - n_2;$

$n_1 = 100, n_2 = 50$

$n_1 = 100 + 50 = 150$

$n_2 = 150 - 50 = 100$

$n_1 = 150 - 100 = 50$

3) Area of circle

double area = Math.PI * radius * radius;

4) Generate 5 random no. b/w 1 to 5

for (int i = 1; i <= 5; i++)

sop(i)(Math.random() * 50));

5) Sum of n numbers

for (int i = 0; i <= n; i++)

{ sum = sum + i;

sop(sum);

int sum = 0;

or sum += i;

2016

14

THURSDAY

APRIL

105-281 • WK 16

0, 1, 1, 2, 3, 5, 8, 13

6) Generate Fibonacci Series: till no. "n"

```
int n1, n2 = 0, n3 = 1;
```

```
for (int i = 1; i <= n; i++)
```

```
{ sop(n3 + " ");
```

```
n1 = n2; n2 = n3;
```

```
n3 = n1 + n2;
```

```
}
```

P.V fibonacci (int n1, int n2, int N)

```
obj.fibonacci(0, 1, 100); { int sum = 0;
```

```
if (n1 == 0)
```

```
sop(n1 + " " + n2);
```

```
sum = n1 + n2;
```

```
if (sum <= N)
```

```
sop(sum);
```

```
n1 = n2;
```

```
n2 = sum;
```

```
fibonacci(n1, n2, N);
```

```
}
```

7) Sum of its digits

```
int sum = 0;
```

```
while (n != 0) or (while n > 0)
```

```
{
```

```
int r = n % 10;
```

```
sum = sum + r;
```

```
n = n / 10;
```

```
}
```

```
sop(sum);
```

8) Factorial of no. "n"

```
int temp = 1;
```

```
while (n > 0)
```

```
{ temp = temp * n;
```

```
n--;
```

```
}
```

```
sop(temp);
```

```
for (int i = n; i > 1; i--)
```

```
{ temp *= i;
```

```
}
```

```
sop(temp);
```

$n(n-1)(n-2) \dots 1$

FRIDAY

APRIL

15

WK 16 • 106-260

2) Reverse the given no. or whether No. is Palindrome

```
int n, r, temp = 0;
```

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

```
sop("Enter num:");
```

```
n = in.nextInt();
```

```
while (n > 0)
```

```
{
```

```
    r = n % 10;
```

```
    temp = temp * 10 + r;
```

```
    n = n / 10;
```

```
}
```

```
sop("Reverse No is: " + temp);
```

```
if (temp == n)
```

→ No. is palindrome

→ which great than 1 &
→ which indivisible by 1 and
itself

→ Always the integer

To find whether a no. is prime or not

Prime No. "n"

smallest prime NO = 2

```
int n, flag = 0
```

```
for (int i = 2; i <= n / 2; i++)
```

```
{ if (n % i == 0)
```

```
    { flag = 1;
```

```
      break;
```

```
    }
```

```
if (flag == 1)
```

```
    sop("Not Prime");
```

```
else
```

```
    sop("Prime");
```

```
public boolean checkPrime(int n)
```

```
{ if (n <= 1)
```

```
    return false;
```

```
for (int i = 2; i <= Math.sqrt(n); i++)
```

```
{ if (n % i == 0)
```

```
    return false;
```

```
}
```

```
return true;
```

```
}
```

2016

16

SATURDAY

APRIL

107-258 • WK 16

1	2	3	4	5	6	7	8	9	10	11	12
14	15	16	17	18	19	20	21	22	23	24	25
28	29	30	31								
M	T	W	T	F	S	S	M	T	W	T	F

9) Program to print Prime No. (say b/w 1 to 100)
1 to n

```

for (int i = 2; i <= n; i++)
{
    if (isPrime(i));
        sop(i);
}

```

```

public static boolean isPrime (int i)
{

```

```

    for (int j = 2; j < i; j++)
    {

```

```

        if (i % j == 0)

```

```

            return false;

```

```

    }

```

```

    return true;
}

```

10) Sorting (in Ascending order) n elements

17 SUNDAY

```

int temp;

```

```

int a[] = {0, 6, 7, 9, 10, 11, 12};

```

```

for (int i = 0; i < n; i++) {

```

```

    for (int j = i + 1; j < n; j++) {

```

```

        if (a[i] > a[j]) {

```

```

            temp = a[i];

```

```

            a[i] = a[j];

```

```

            a[j] = temp;

```

```

        }
    }
}

```

2016

MONDAY

APRIL

18

sop("Ascending order is: ");

for (int k=0; k<n; k++)

~~sop(a[k]);~~

sop(a[k] + ",");

(ii) Take input from user & store in array:

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

Scanner in = new Scanner(System.in);

sop("Enter no. of elements: ");

int n = in.nextInt();

int a[] = new int[n];

for (int i=0; i<n; i++)

{ sop("Enter element: ");

a[i] = in.nextInt();

}

(12) Armstrong Number $153 = 1^3 + 5^3 + 3^3 = 153$

int n = 153, r = 0, x;

~~int~~

while (n > 0)

{ r = n % 10;

x = x + (r * r * r)

n = n / 10;

}

if (x == n)

else

→ Not Armstrong

Math.pow(r, 3)

2016

19

TUESDAY

APRIL

110-256 • WK 17

array.length
string.length()

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
28	29	30	31									
M	T	W	T	F	S	S	M	T	W	T	F	S

13) WAP To find largest & smallest no. in an Array:

```

int ar[] = new int[] {55, 65, 45, 95, 75, 3};
int smallest = ar[0];
int largest = ar[0];
for (int i = 1; i < ar.length; i++) {
    if (ar[i] > largest)
        largest = ar[i];
    else if (ar[i] < smallest)
        smallest = ar[i];
}
sop(smallest);
sop(largest);

```

14) Array Comparison : (size & content)

```

p.s. booleaneq(int a[], int b[]) {
    if (a.length != b.length) return false;
    for (int i = 0; i < a.length; i++) {
        if (a[i] != b[i]) return false;
    }
    return true;
}

```

psvm(s...a)

2016

```

int[] a = {3, 1, 4, 1, 5};
int[] b = {3, 1, 4, 1, 3};

```

```

sop(eq(a, a)); // T
sop(eq(a, b)); // F

```


WEDNESDAY

APRIL

20

WK 17 • 111-255

15) String Comparison:

```
Scanner in = new Scanner(System.in);  
sop("enter s1:");
```

```
String s1 = in.nextLine();  
sop("enter s2:");
```

```
String s2 = in.nextLine();
```

```
if (s1.equals(s2))
```

```
if (s1.contentEquals(s2))
```

→ equals

```
else
```

→ not equal

16) String - Reverse

```
String temp = "";
```

```
String s = "ABC";
```

```
for (int i = s.length() - 1; i >= 0; i--)
```

```
temp = temp + s.charAt(i);
```

```
sop(temp);
```

17) String Case Conversion

```
String s1 = "HELLO";
```

```
String s2 = "world";
```

```
String l = s1.toLowerCase();
```

```
String u = s2.toUpperCase();
```

```
sop(l);
```

```
sop(u);
```

2016

21

THURSDAY

APRIL

112-254 • WK 17

W → To move cursor
 \t → To put tab (space)

18) String Substring

String s1 = "Tester";

```
String str2 = s1.substring(0,4);
sop(str2);
```

19) WAP to print IP Address of the System.

```
sop(InetAddress.getLocalHost());
```

20) Power of a no.

```
sop(Math.pow(2,3));
```

```
sop(Math.pow(5,2));
```

Patterns

①

*

* *

* * *

* * * *

* * * * *

rows → 5

↓
IP can be taken from user

```
for (int i = 1; i <= 5; i++)
```

```
{
  for (int j = 1; j <= i; j++)
```

```
{
  sop("* ");
```

```
System.out.print("\n");
```

```
System.out.println();
}
```

2016

cursor down
(space) line

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	
S	M	T	W	T	F	S	S

FRIDAY

APRIL

22

WK 17 • 113-253

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5

```

for ( — )
{
    for ( — )
    {
        print(i);
    }
    print("\n");
}
    
```

row = 5;

```

int num = 1;
for (int i = 1; i <= 5; i++)
{
    for (int j = 1; j <= i; j++)
    {
        print(num + " ");
        num++;
    }
    print("\n");
}
    
```

1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

4

```

for ( — )
{
    int num = 1;
    for ( — )
    {
        print(num);
    }
}
    
```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5

5

```

int k = 1;
for ( — )
{
    for (int j = 1; j <= k; j++)
    {
        print("*");
    }
    K = K + 2;
    print("\n");
}
    
```

* * *
* * * * *
* * * * *
* * * * *
2016

23

SATURDAY

APRIL

114-252 • WK 17

space = rows - 1

1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26
28	29	30	31									
M	T	W	T	F	S	S	M	T	W	T	F	S

row = cols

row = 5

```

6 int space = 4;
  int i, j;
  for (int i = 1; i <= 5; i++)

```

```

    {
      for (int j = 1; j <= space; j++)
      {
        print(" ");
      }

```

space = space - 1;

for (int j = 1; j <= i; j++)

{ print("*");

{ print("\n");

```

7 int space = 6, K = 1;
  int i, j;
  for (int i = 1; i <= 5; i++)

```

{

for (int j = 1; j <= space; j++)

{ print(" ");

{

24 SUNDAY

space = space - 2;

for (j = 1; j <= K; j++)

{ print("*");

{

K = K + 2;

2016

print("\n");

```

      *
    * *
  * * *
* * * *

```

```

      *
    * * *
  * * * *
* * * * *

```

MONDAY

APRIL

25

```
6) int i, j row = 5;
```

$$for(i=1; i \leq row; i++)$$

```
for (j=1; j<=row-1; j++)
    print (" ");
```

```
print (" ");
```

```
for (j=1; j<=i; j++)
```

```
print("*").
```

```
for (j = i - 1; j >= 1; j--) {
    print ("*");
}
```

print ("x");

```
println();
```

2

→ 1st time, it won't print

2nd time $\rightarrow x$

3rd time \rightarrow ***

26

TUESDAY
APRIL

Delete duplicate element in an array

int a[] = {10, 20, 30, 20, 40, 40, 50, 60, 70, 80};

int size = a.length;

sop("Before" + size); // 10

for (int i = 0; i < size; i++)

{
for (int j = i + 1; j < size; j++){
if (a[i] == a[j]){
while (j < (size) - 1){
a[j] = a[j + 1];

j++

size--;

}

sop("After" + size); // 8

for (int k = 0; k < size; k++)

sop(a[k]);

1 2 3 4 5 6 7 8
 9 10 11 12 13 14 15 16 17 18 19 20 21 22
 23 24 25 26 27 28 29 30 31
 M T W T F S S M T W T F S S

WEDNESDAY

APRIL

27

WK 18 • 118-248

→ Delete duplicate characters from a word/string.

String s = "abcabcabc";

char ch;

String result = "";

for (int i = 0; i < s.length(); i++)

{

ch = s.charAt(i);

if (ch != ' ') → space

result = result + ch;

s = s.replace(ch, ' '); → Replace all occurrence of the current char by a space.

System.out.println("After " + result); // abc