Java pre-programming

Tuesday, December 10, 2024 7:21 AM

Duration 30-35 days.

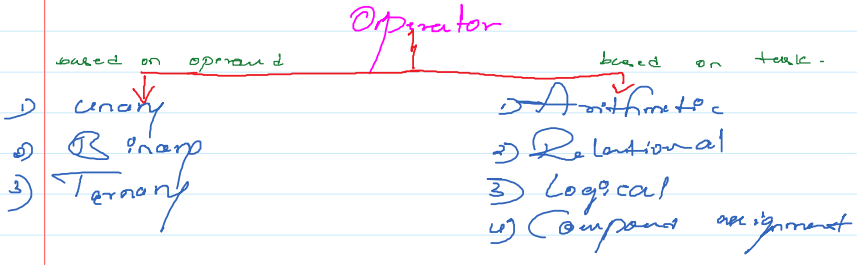
1) Basic
2) Number program
3) Pattern

——— final mock ———→

4) Array Programs
5) String Programs.

———→ final mock ———→

Operators :- Pre-defined Symbols used to perform some specific task.



Unary :- It will accept only one operand.

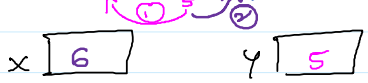
ex:- Increment operator (++)
Decrement operator (--)
typecast operator ((type))
new operator
not operator (!)

1) Increment operator.

x 56

ex:- x = 5

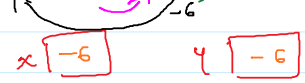
y = x++;



x -6

x = -7

y = ++x;



x 345

x = 3

y = ++x + x++

$x = 3$

$y = ++x + x++$

$y = 4 + 4$

$y = 8$

x | 5 | y | 8 |

x | 1 2 3 4 5 |

$x = 1$

$y = x++ + ++x + x++ + x++;$

$y = 1 + 3 + 3 + 4$

$y = 11$

x | 5 | y | 11 |

x | 8 9 8 7 8 7 |

$x = 8$

$y = x++ + --x + x-- + ++x - --x$

$y = 8 + 8 + 8 + 8 - 7$

$y = 32 - 7$

$y = 25$

x | 7 | y | 25 |

x | 3 4 3 2 3 2 |

$x = 2$

$y = 4$

$x = 3$

$y = x++ + --x * x-- - ++x + --x;$

$y = 3 + 3 * 3 - 3 + 2$

$y = 9 + 2$

$y = 11$

x | 2 | y | 11 |

1) + (plus) If any one of the operand is String. plus(+) will act as concat operator.

1) 10 + 20 => 30

2) 'a' + 10
97 + 10
=> 107

3) 10 + 'a' + 2
10 + 97
107 + 2
=> 109

4) "hello" + "hi" (concat)
"hellohi"

5) 'a' + "hello" (concat)
"ahello"

6) 10 + 20 + "hello"
30 + "hello" (concat)
"30hello"

7) "hi" + 10 + 20
"hi10" + 20
"hi1020"

1) 'z' / 'a'

2) 'a' % 2

3) 'a' / 2

1) Tom wants to buy a Tshirt. Tshirt size should be small and price should be less than or equals to 1000.

1) size = 'M'
price = 800
X

2) size = 'S'
price = 1200
X

3) size = 'S'
price = 950
✓
eligible

1) char size = 'M';
int price = 800;

2) size = 'S'
price = 950;

size == 'S' && price <= 1000
'M' == 'S' X — not eligible
'S' == 'S' ✓ && 950 <= 1000 ✓

's' == 's' && 950 <= 1000
Eligible

Smith wants to buy a Tshirt,
Tshirt color should be red or black

color = "green"

color = red

color = black

Color = "green";

color = "black"

Color == "red" || color == "black"
"green" == "red" X || "green" == "black" X not elig.
"black" == "red" X || "black" == "black" ✓
Eligible.

Small

medium

large

nextInt()
nextByte
nextBool

nextLine()

nextInt().charAt(0)

next()
nextLine()

Size = M
Price = 800

) WAP Check Person is Eligible to vote or not.

int age = 21;

(age >= 18)

int age = 21;

age >= 18
Eligible

2) WAJP check given value is Even or odd.

int num = 9;

num % 2 == 0
Even

WAJP Print Odd no blw the Ran
WAJP print Even no. blw the Range.
(1 to 10)

1 2 3 4 5 6 7 8 9 10

n

n % 2 == 0 sopln(n)

1 % 2 == 0 X
2 % 2 == 0 ✓
3 X
4 ✓
5 X
6 ✓
7 X
8 % 2 == 0 ✓
9 X
10 ✓

2
4
6
8
10

WAJP Find Sum of Even number blw the range;

Sum = 0;

-7 1 2 3 4 5 6 7 8 9 10

i

i % 2 == 0 sum = sum + i;

1 % 2 == 0 X
2 % 2 == 0 ✓ 2 = 0 + 2
3 % 2 == 0 X
4 % 2 == 0 ✓ 6 = 2 + 4
5 % 2 == 0 X
6 % 2 == 0 ✓ 12 = 6 + 6
7 % 2 == 0 X
8 % 2 == 0 ✓ 20 = 12 + 8
9 % 2 == 0 X

7%2==0 ✗
8%2==0 ✓
9%2==0 ✗
10%2==0 ✓

20 = 12 + 8

30 = 20 + 10

sop("sum is" + sum); // SO

WAJP Find Sum of first 10 even number (1 to 50)

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 .....

50

(10 → 100) Sum of first 5 even no.

10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 100

10 + 12 + 14 + 16 + 18 → 70

WAJP count positive even number between the range. (-10 to 10)

i = -10 i <= 10

-10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

X x

o/p -> 5

n = 32948 ;

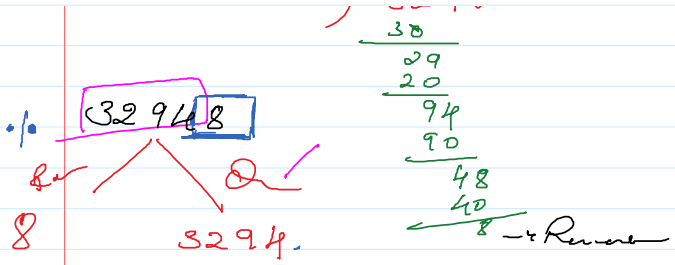
Divide given no. with 10

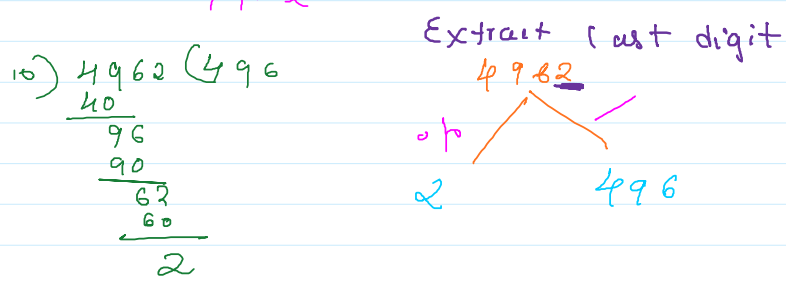
n%10 n/10

Remainder Quotient

10) 32948 (3294 → Quotient

38





int rem = n%10

4962%10

int rem = 2

Remove last digit

4962

o/p

2

496 → Quotient

int q = n/10

4962/10

int q = 496

KCAISP Print multiplication table for given no.

int n=5;

o/p →

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 10 = 50

WAJP print Ascii value A -> Z.

op :- A -> 65
B -> 66
C -> 67
.
.
.
Z -> 90

ch = A ; ch <= Z

loop
Initialization
condition
updation

WAJP Extract all digit from given value:

int n = 9268;

	Extract last digit	print rem	Remove last digit
n > 0	int rem = n % 10	sop(rem)	n = n / 10
1st iteration 9268 > 0 ✓	9268 % 10	8	= 9268 / 10
	rem = 8		n = 926
2nd iteration 926 > 0 ✓	= 926 % 10	6	n = 926 / 10
	rem = 6		n = 92
92 > 0 ✓	rem = 92 % 10	2	= 92 / 10
	rem = 2		n = 9
9 > 0 ✓	rem = 9 % 10	9	n = 9 / 10
	rem = 9		n = 0
0 > 0 ✗ loop terminate			

10 | 9268 (926
90
26
20
68
60
8 R

n / 10

10 | 9 (0 -> 9
0
9 -> R

Note :- Removing digit will act as updation part for loop

WAJP count no. of digit in given no.

int n = 5839474; o/p -> 5

1st iteration 5839474 > 0 ✓ cnt = 1

int n = 5439; o/p -> 5

		ctr
1st	5439 ✗	1
2nd	543 ✗	2
3rd	54 ✗	3
4th	5 ✗	4
5th	✗	5

o/p.

6th time x-> loop terminate.

Task

1) WAP print only even digit in given no.
2) WAP print only odd digit " "
3) WAP find sum of each digit in given no.
4) WAP find sum of digit, if it contain 1 (or) else print no. as it is.
5) WAP find odd digit sum.

i/p -> 529 i/p 519
o/p -> 529 o/p sum is : 15

4) WAP find sum of digit, if it contain 1 (or) else print no. as it is.

int n = 5912; boolean flag = ~~false~~ true;

5912 rem
↳ 2 == 1 ✗

5912
↳ 1 == 1 ✓ flag = true; break; loop termin

if (flag == true)
{
  sum of each digit
}
else
{
  sop(n);
}

```
package Number_Program;

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

```
package Number_Program;

public class Sum_of_Digit {
public static void main(String[] args) {
int n=5624;
int sum=0;
while(n>0) {
int rem=n%10;
```

```java
package Number_Program;

public class Extract_All_Digit {
public static void main(String[] args) {
int n=452;
while(n>0) {
int rem=n%10;
System.out.println(rem);
n=n/10;
}
}
}
```

```java
package Number_Program;

public class Count_Digit {
public static void main(String[] args) {
int n=54398;
int c=0;
while(n>0) {
n=n/10;
c++;
}

System.out.println("no.of digits: "+c);
System.out.println("_____");
int c1=0;
for(int i=342;i>0;i=i/10) {
c1++;
}
System.out.println("no.of digits: "+c1);
}
}
```

```java
package Number_Program;

public class Even_digit {
public static void main(String[] args) {
int n=547821;
while(n>0) {
int rem=n%10;
if(rem%2==0) {
System.out.println(rem);
}
n=n/10;
}
}
}
```

```java
package Number_Program;

public class Odd_Digit {
public static void main(String[] args) {
int n=547821;
while(n>0) {
int rem=n%10;
if(rem%2!=0) {
System.out.println(rem);
}
n=n/10;
}
}
}
```

```java
public static void main(String[] args) {
int n=5624;
int sum=0;
while(n>0) {
int rem=n%10;
sum=sum+rem;
n=n/10;
}
System.out.println("sum is: "+sum);
}
}
```

```java
package Number_Program;

public class Sum_digit_if_it_contain_1 {
public static void main(String[] args) {
int n=64132,n1=n;
int sum=0;boolean flag=false;
while(n>0) {
int rem=n%10;
if(rem==1) {
flag=true;break;
}
n=n/10;
}
if(flag) {
//find sum of each digit
while(n1>0) {
int rem=n1%10;
sum=sum+rem;
n1=n1/10;
}
System.out.println("sum is: "+sum);
}
else {
System.out.println("Number is: "+n1);
}
}
}
```

WAP check given number is Spy no. or not.

Spy Number :- Sum of each digit is equals to product of each digit.

int n = 532;

5+3+2 = 10 sum == pro

int n = 532;

5+3+2 = 10 sum == pro

5*3*2 = 30 10 == 30

not a spy no.

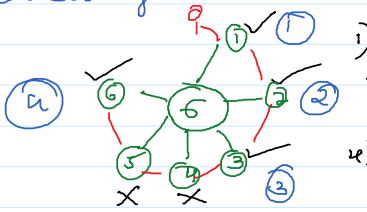
int n=123;

1+2+3 = 6 sum == pro

1*2*3 = 6 6 == 6 ✓ spy no.

WAP find factors of given value.

factors :- Factors are the numbers, which divides given number completly

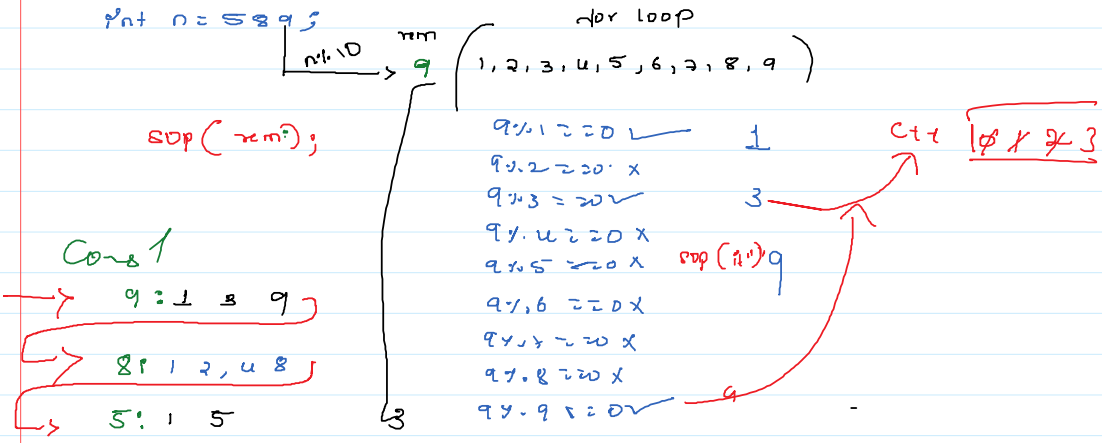


1) 6 (6 2) 6 (3 3) 6 (2
6 6 6
0 0 0

4) 6 (1 5) 6 (5 6) 6 (6
4 5 6
2 1 0

WAP find Factor of each digit in given no.

int n = 589;



soplin( ) : to shift cursor to next line

WAP Count no. of factors.

int n=532;

WAP Check given no. is Prime or not.

Prime no :- Any number which is divisible by only 1 and itself. Number must contain exact 2 different factors.

int n = 5; c = 0;

5 % 1 == 0 ✓ ctr [1]
5 % 2 == 0 ✗
5 % 3 == 0 ✗
5 % 4 == 0 ✗
5 % 5 == 0 ✓ ctr [2]

6 times loop for

c == 2
2 == 2
prime no.

int n = 4, c = 0;

4 % 1 == 0 ✓ ctr
4 % 2 == 0 ✓ ctr
4 % 3 == 0 ✗
4 % 4 == 0 ✓ ctr

c == 3
3 == 2 ✗
not a prime no.

WAP print prime number blw the range.
(1 to 10)

1 2 3 4 5 6 7 8 9 10
n

to traverse range we require loop.

```
for(int n = 1; n<=10; n++)
{
    c = 0;
    // count factors.
    for(int i = 1; i<=n; i++)
    {
    }
    (c == 2).
    {
        cout(n)
    }
}
```

n = 1 2

WAP Reverse number.

int n = 123;

rev = rev * 10 + rem
= 0 * 10 + 3
rev = 0 + 3
rev = 3

= 3 * 10 + 2
= 30 + 2
rev = 32

rev = 32 * 10 + 1
= 320 + 1
rev = 321

321

WAP check given number is Palindrome or not.

Palindrome :- Given number must be equals to it's reversed no.

int n = 121; o/p → 121

given no == reversed no

121 == 121 ✓

→ Palindrome.

WAP Find factorial of given value.

Factorial :- Product of all positive integer start from 1 to till given no.

int n = 5;

1 x 2 x 3 x 4 x 5

= 120

int n = 4 :> 1 x 2 x 3 x 4

= 24

WAP check given number is Spy no. or not.

Spy Number :- Sum of each digit is equals to product of each digit.

int n = 532;

5+3+2 = 10

5 * 3 * 2 = 30 sum == pro

10 == 30 not a spy no.

int n = 123;

1+2+3 = 6 sum == pro

1 * 2 * 3 = 6 6 == 6 ✓ spy no.

(1 to 500)

int n = 22; o/p.

sum = 0

pro = 1

{ while

sum = sum + rem

pro = pro * rem

}

sum == pro

sum == prd
{
sqr(n)
}

Inp: Check given no. is Perfect number or not

Perfect No :- Sum of factors except that number is equals to given value.

int n = 8;

8: 1 2 4 ~~8~~

1+2+4

= 7

n == sum

8 == 7 ✗

Not a Perfect no

int n = 6

6: 1 2 3 ~~6~~

1+2+3

Sum: 6

n == sum

6 == 6 ✓ Perfect no.

Inp: Find factorial for each digit

int n = 452; rem (Find factorial for 2) → factorial prg

Split(rem → fact)

2 → 2

5 → 120

4 → 24

n = 452