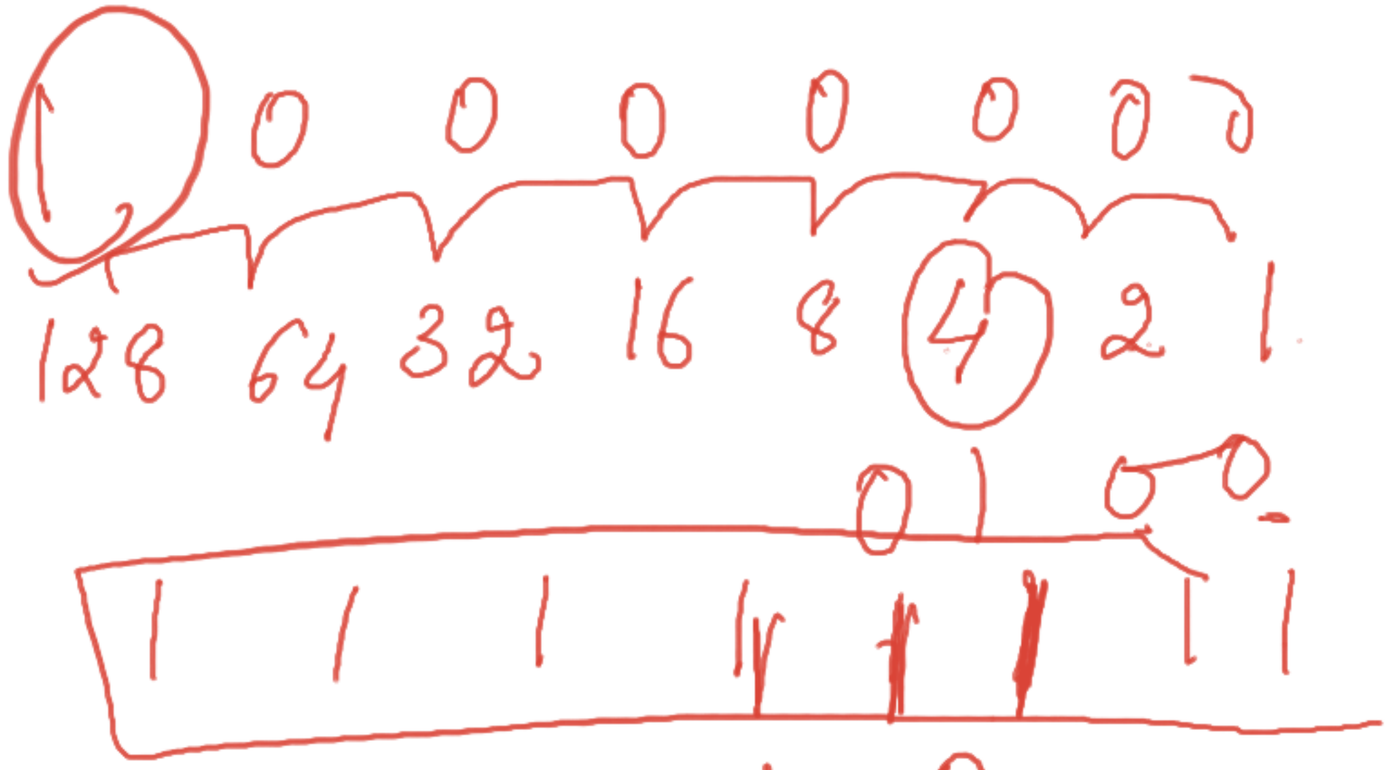
1. Syso ("Value of a and b is" + 10 + 20); → 30X → 1020 → concatenate opn

2. int a = 20;
   int b = 30;
   Syso (a+b + "Sum is"); → 50 Sum is ✓

3. int x = 100;
   int y = 20;
   Syso ("Sum is" + (x+y)); → addition operation
   Sum is 120;

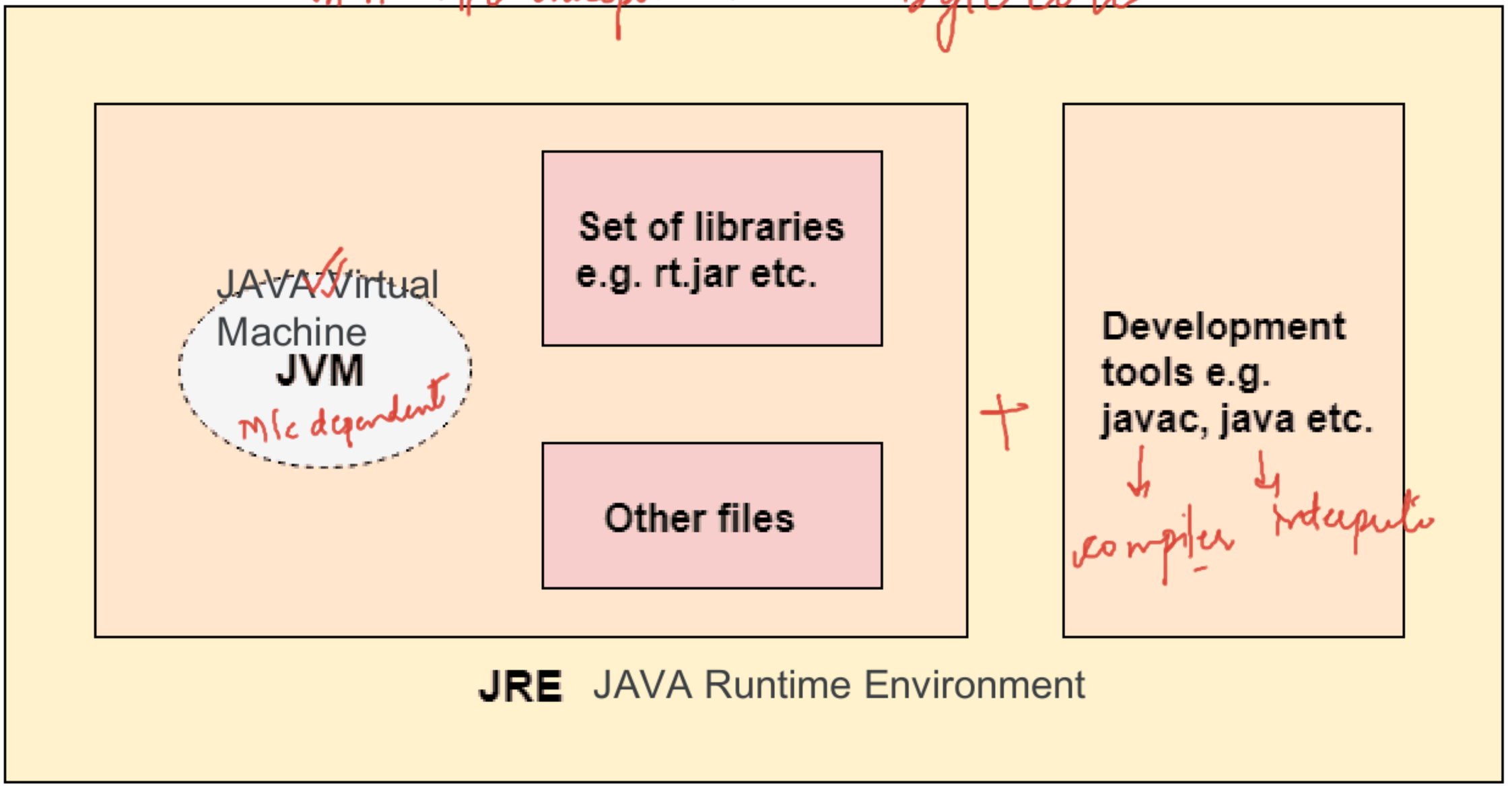
4. int z = 20;
   z++; 21
   ++z; 22
   Syso (z); // 22 ✓ → 24 → M/m
   Syso (z++ + ++z); // 22 + 24 → 46
   Syso (z); // 24 ✓

Addition



4 Binary → 0 1 0 0 → 0 - 127 → 128

JAVA - M/c independent → Byte code



JRE JAVA Runtime Environment

JAVA Developemnt KIT **JDK** → MAC, LINUX, Windows

JAVA S/W Run → JDK ? or JRE ?

development

if you don't want to develop the code, only execution

Precedence	Operator	Operand type	Description
1	++,	Arithmetic	Increment and decrement
1	+, -	Arithmetic	Unary plus and minus
1	~	Integral	Bitwise complement
1	!	Boolean	Logical complement
1	(*type*)	Any	Cast
2	*, /, %	Arithmetic	Multiplication, division, remainder
3	+, -	Arithmetic	Addition and subtraction
3	+	String	String concatenation
4	<<	Integral	Left shift
4	>>	Integral	Right shift with sign extension
4	>>>	Integral	Right shift with no extension
5	<, <=, >, >=	Arithmetic	Numeric comparison
5	instanceof	Object	Type comparison
6	==, !=	Primitive	Equality and inequality of value
6	==, !=	Object	Equality and inequality of reference
7	&	Integral	Bitwise AND
7	&	Boolean	Boolean AND
8	^	Integral	Bitwise XOR
8	^	Boolean	Boolean XOR
9	|	Integral	Bitwise OR
9	|	Boolean	Boolean OR
10	&&	Boolean	Conditional AND
11	||	Boolean	Conditional OR
12	?:	N/A	Conditional ternary operator
13	=	Any	Assignment

String

+

Concatination

"Testing" + (123 + 458)

addition

Testing 123458

"Sum of" + a + "and" + b

value

Arithmetic op

+

addition

123 + 1 + "String"

124 String

/

divide → Q

4/2

2 → Q

2 ⟌ 4

4

0 → Rem.

%

Modulous → R

4 % 2

2 ⟌ 4

4

0 → R

5

2 ⟌ 11

10

1 → 1

11 / 2

5 * 5 X X

2 ⟌ 11

10

10

10

0

int s = 11/2

s = 5

doubled = 5.0

1. whether Num is Even or odd?
2. reverse of a number?
3.number is pallindrom or not?
4. Armstrong or not?
5. Swapping of number

1. Swapping of a number

a = 12

b = 4

---

a = 4

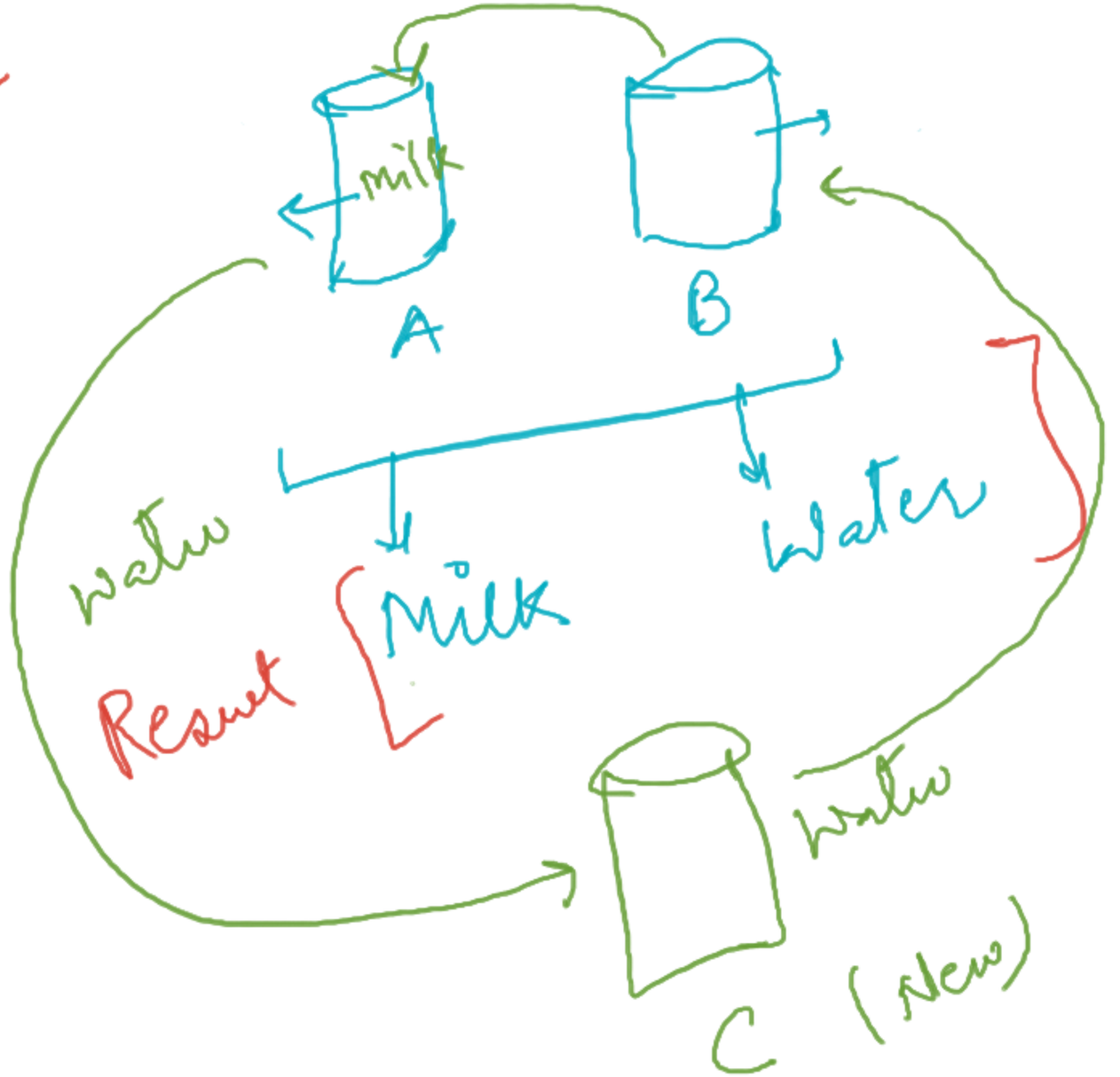
b = 12

```
int c;
c = a;
a = b;
b = c
```

4 ← c = a;

12 ← a = b;

4 ← b = c



Reverse of Number :→

→ 21

→ O/P → 12

133 → 331

467 → 764

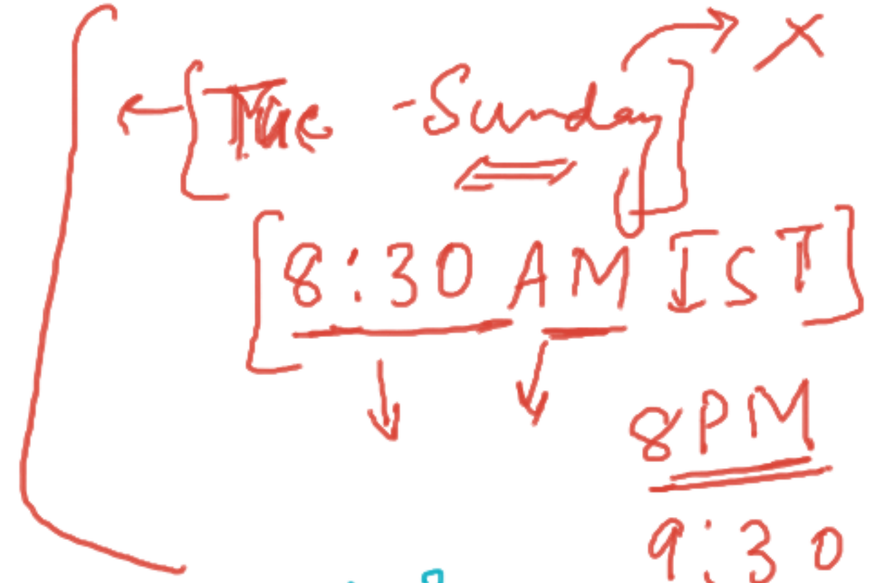
Logic

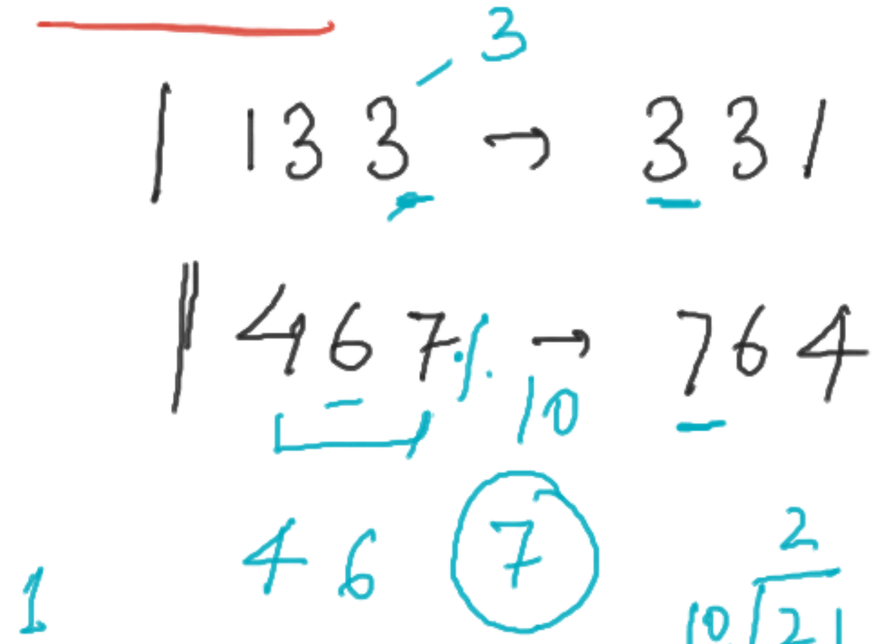
St. L. 21 % 10 → 1

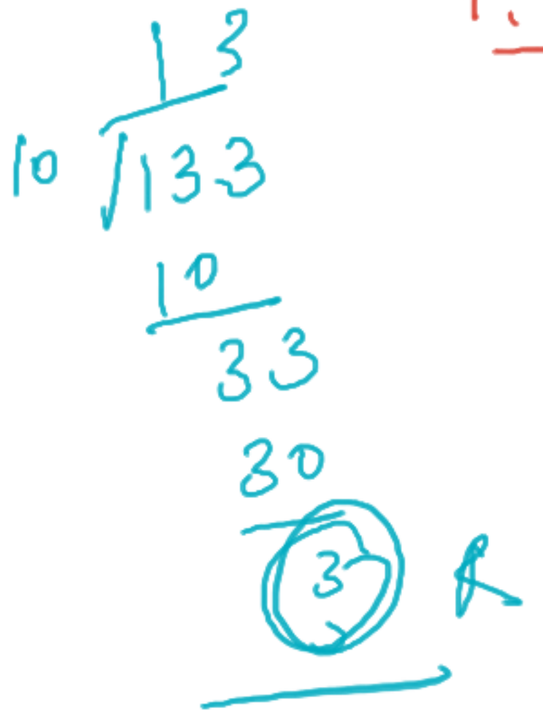
fd = 21 % 10 ; // 1

fd 21 / 10 ; // 2

"No. of" T. ld + fd







Sl no. r1 = 234 % 10 → 4

n = 234 / 10 → 23

second digit

r2 = n % 10 → 3 ✓

r3 = n / 10 → 2 (Rev of No is r1 + r2 + r3)

23 % 10 → 3

(1) 23 → Q

10 | 234 → 23

20

34

30

4 → R

% 10

LearnJava - JAVA_Demos/src/operators/programs/ReverseNumber.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- JAVA_Demos
  - JRE System Library [JavaSE-15]
  - src
    - (default package)
      - VariablesDemo.java
      - VariablesEx2.java
    - DataTypes
    - operators
      - ArithemeticOpt.java
      - PreIncrementDecDemo.java
      - UniaryOperatorDemo.java
    - operators.programs
      - ReverseNumber.java
      - SwapNumbers.java

ExplicitTypeCasting.java | ArithemeticOpt.java | VariablesEx2.java | VariablesDemo.java | SwapNumbers.java | ReverseNumber.java


```
1  package
2
3  public class ReverseNumber {
4
5      public static void main(String[] args) {
6          int n= 456;
7
8          int rem1 = n % 10; // 172%10-->2
9
10         n = n/10; // 172/10 ---> 17
11         int rem2 = n%10;// 17%10-->7
12
13         int rem3 = n/10;
14
15
16
17         System.out.println("Reverse of a number is-"+ rem1+rem2+rem3);
18
19
20     }
21
22 }
23
```

Writable | Smart Insert | 8 : 41 [10]