**Built-in Packages:**

These packages consist of a large number of classes which are a part of Java API. Some of the commonly used built-in packages are:

1) **java.lang:** Contains language support classes(e.g classed which defines primitive data types, math operations java.lang.math). This package is automatically imported. Eg: byte, boolean, float, enum, int etc.

2)  **java.io:** Contains classed for supporting input / output operations, all file operations, streams etc.

3)  **java.util:** Contains utility classes which implement data structures like Linked List, Dictionary and support; for Date / Time operations.

Eg: arrays, calenders, collections, date, hashmap, dictionary, enum, maps , Scanner, stringTokenizer etc

4)  **java.applet:** Contains classes for creating Applets.

5)  **java.awt:** Contain classes for implementing the components for graphical user interfaces (like button , ;menus etc).

6)  **java.net:** Contain classes for supporting networking operations.

7)  **javax.swing:** Java Swing is a lightweight Graphical User Interface (GUI) toolkit that includes a rich set of widgets. Eg: Jlabel, Jtext, JList etc.

There are six numeric types four integer and two floating point:

* byte 1 byte -128 to 127
* short 2 bytes -32,768 to 32,767
* int 4 bytes -2,147,483,648 to 2,147,483,647
* long 8 bytes -9,223,372,036,854,775,808 to 9,223,372,036,854,775,80
* float 4 bytes 7 decimal digits
* double 8 bytes 16 decimal digits
* Char 2 bytes 65,536(unsigned)
* Boolean true or false

|  |  |  |
| --- | --- | --- |
| **Operator Type** | **Category** | **Precedence** |
| Unary | postfix | *expr*++ *expr*-- |
| prefix | ++*expr* --*expr* +*expr* -*expr* ~ ! |
| Arithmetic | multiplicative | \* / % |
| additive | + - |
| Shift | shift | << >> >>> |
| Relational | comparison | < > <= >= instanceof |
| equality | == != |
| Bitwise | bitwise AND | & |
| bitwise exclusive OR | ^ |
| bitwise inclusive OR | | |
| Logical | logical AND | && |
| logical OR | || |
| Ternary | ternary | ? : |
| Assignment | assignment | = += -= \*= /= %= &= ^= |= <<= >>= >>>= |

**Sample programs:**

Ex1:

int x=10;

x++;

sopln(x); //x=11 post increment

Ex2:

int a=10,b=20,c=30;

b=a;c=b;

System.out.println(c); //10

Int a=b=c=10; //a already declared

System.out.println(c); //10

//int a,b,c; //a, b, c values already declared

a=b=c=10;

System.out.println(c); //10

Ex3:

char ch='a';

ch++;

int a=ch;

System.out.println(ch+" "+a); //b 98

Ex4:

double d=10.5;

d++;

System.out.println(d+" "+(d++)+" "+(++d)); //left to right evaluating

int a=10;

System.out.println(++a+" "+(a++)+" "+(a--)+" "+(--a)); //left to right evaluating

Output:

11.5 11.5 13.5

11 11 12 10

Ex5:

boolean b=true;

b++;

output:

bad operand type boolean for unary operator '++'

Ex6:

byte b=20;

byte b=b+1; //b already defined

//lossy conversion from int to byte

//**overflow** - where the result of a calculation exceeds the ability of the allocated memory location to be able to store it.

byte b=(byte)b+1 ;

System.out.println(b);

Ex7:

byte a=10;

byte b=20;

byte c=a+b;

byte c=byte(a+b);

sopln(c);

output:

lossy conversion from int to byte

**overflow** - where the result of a calculation exceeds the ability of the allocated memory location to be able to store it.

Ex8:

System.out.println(10/0);

System.out.println(10/0.0);

output:

Throwing arithmetic exception

Ex9:

System.out.println('a' + 'b'); //195

System.out.println('a' + 1); //98

System.out.println('a' + 1.2); //98.2

Ex10:

String a="ashok";

int b=10 , c=20 , d=30 ;

a=b+c+d ; //int cannot converted into string

a=a+b+c ;

b=a+c+d ; //string cannot converted into int

Ex11:

System.out.println(10 < 10.5);

System.out.println('a' > 100.5);

System.out.println('b' > 'a');

System.out.println(true != false); //bad operand types for operator '>'

Output:

True

False

True

True

Ex12:

System.out.println(10 == 20) ; //false

System.out.println('a' == 'b' ); //false

System.out.println('a' == 97.0 ); //true

System.out.println(false == false); //true

Ex13:

Thread t1=new Thread( ) ;

Thread t2=new Thread( );

Thread t3=t1 ; //thread t1 and t3 performing same task

System.out.println(t1==t2);

System.out.println(t1==t3);

Output:

False

True

Ex14:

System.out.println(true&false); //true, if all inputs are true

System.out.println(true|false); //false, if all inputs are false

System.out.println(true^false); //if all cases are either true or false then only it is false

Output:

False

True

True

Ex15:

System.out.println(4&5); //4

System.out.println(4|5); //5

System.out.println(4^5); //1

//converts numbers into bits, and then performs logical operations

Ex16:

System.out.println(~true);

System.out.println(~4); //-5

Output:

~10=-11

Ex17:

System.out.println(!true); //false

System.out.println(!4); //bad operand type int for operator !

Ex18:

int x=10 , y=15 ;

if(++x < 10 || ++y > 15) { //instead of || using &,&&, | operators

x++;

}

else

y++;

System.out.println(x+"----"+y);

Output:

12…..16 for ||

11…..16 for &&

11 ….17 for &

12….16 for |

Ex19:

int x=130;

byte b=(byte)x;

System.out.println(b);

Output:

-126 //it is recycling after exceeds maximum value

//251 …….-5

//135 ……-121

Ex20:

int x=150;

short s=(short)x; //150

byte b=(byte)x; //-106

System.out.println(s);

System.out.println(b);

// byte giving total of 256 value

Ex21:

double d=130.456 ;

int x=(int)d ;

System.out.println(x); //130

byte b=(byte)d ;

System.out.println(b); //-126

Ex22:

int x=(10>20)?30:((40>50)?60:70);

System.out.println(x); //70

Ex23:

class OperatorsDemo {

public static void main(String[] args) {

sopln(m1(1)+m1(2)\*m1(3)/m1(4)\*m1(5)+m1(6));

}

public static int m1(int i) {

sopln(i);

return i;

}

}

Output:

1

2

3

4

5

6

12