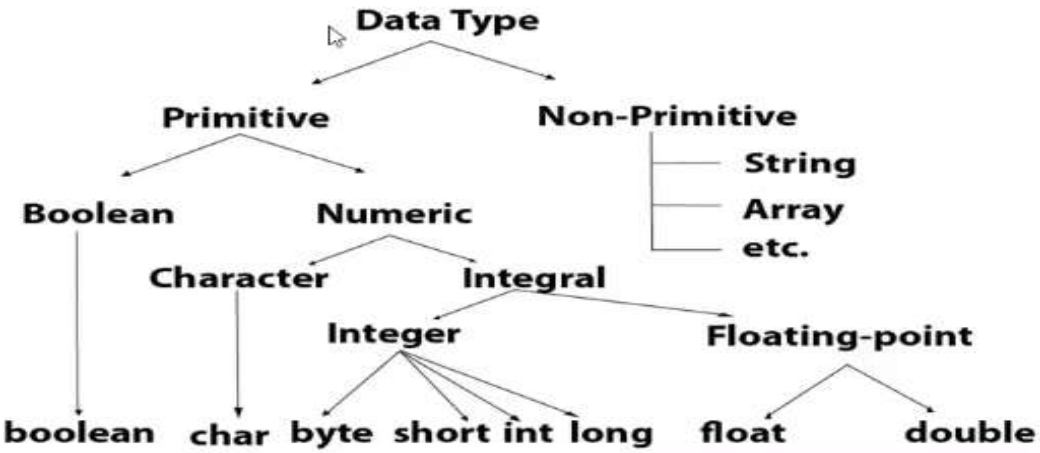
Basics

11 October 2021 10:23 AM

We use the static void main bcoz it is a starting point as it's a void main, but static we use bcoz, Whichever class is static class javac loads those into memories.

We use "**System.out.print();"** bcoz in java there is a outstream file which has print commands.
Use **"System.out.println("hello");" --> like "(hello\n")** so every time it prints hello, compiler goes to next line.

DATA TYPES:



Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte
byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

Range of numeric data types in Java

Type	Size	Range
byte	8 bits	-128 .. 127
short	16 bits	-32,768 .. 32,767
int	32 bits	-2,147,483,648 .. 2,147,483,647
long	64 bits	-9,223,372,036,854,775,808 .. 9,223,372,036,854,775,807
float	32 bits	$3.40282347 \times 10^{38}$, $1.40239846 \times 10^{-45}$
double	64 bits	$1.7976931348623157 \times 10^{308}$, $4.9406564584124654 \times 10^{-324}$

```
public class Hello{
        public static void main(String ar[]){
                int a= 10;
                int b = 20;
                int c = 30;
                if(a>b){
                        if(a>c){
                                System.out.println(a);
                        }else{
                                System.out.println(c);
                        }
                }else if(b>c){
                        System.out.println(b);
                }else{
                        System.out.println(c);
                }
        }
}
```

do-while loop is a exit control loop bcoz it will first do the operation and then check the condition.
While loop is entry control loop
For loop is entry control loop

Prime number

```
public class PrimeExample{
 public static void main(String args[]){
  int i,m=0,f=0;
  int n=3;
  m=n/2;
  if(n==0||n==1){
   System.out.println(n+" is not prime number");
  }else{
   for(i=2;i<=m;i++){
    if(n%i==0){
     System.out.println(n+" is not prime number");
     f=1;
     break;
    }
   }
   if(f==0)  { System.out.println(n+" is prime number"); }
  }
```

```
}
}
```

```
3 is prime number
```

Fibonacii series.

```
public class fibo{
	public static void main(String ar[]){
		int a = 0;
		int b = 1;
		int c;
		System.out.println(a);
		System.out.println(b);
		for( int i = 0 ; i < 8 ; ++i)
		{
			c = a + b;
			System.out.println(c);
			a = b;
			b = c;
		}
	}
}
```

```
0
1
1
2
3
5
8
13
21
34
```

Factorial of a number

```
public class fact{
	public static void main(String ar[]){
		int n = 10;
		int fact = 1;
		for( int i = 1 ; i < n ; i ++ ){
			fact = fact * i;

		}
	   System.out.print(fact);
	}
}
```

```
362880
```

Armstrong number For e.g. 153: 1*1*1 + 5*5*5 + 3*3*3 = 153

```
public class arm{
	public static void main(String ar[]){
		int a = 153;
		int a1 = a;
		double sum;
		for(  sum = 0 ; a != 0 ; a/=10){
		  int m = a % 10;
		  double num = Math.pow(m,3);
		  sum = sum + num;
		}
		if(sum == a1){
		  System.out.print(a1 + " is Armstrong Number");  }
	}
}
```

```
153 is Armstrong Number
```

Sum of digits

```
public class sum{
	public static void main(String ar[]){
		int a = 154;
		int sum;
		for(  sum = 0 ; a != 0 ; a/=10){
		  int m = a % 10;
		  sum = sum + m;
		}
		System.out.println("Sum of digits is "+sum);
	}
}
```

```
Sum of digits is 10
```

```
public class pattern2{
	public static void main(String ar[]){
		for( int i = 0 ; i <= 10 ; i ++){
			for(int j = 1 ; j <= i; j++){
				System.out.print(j+" ");
			}
		System.out.println(" ");
		}
	}
}
```

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

```
public class pattern1{
	public static void main(String ar[]){
		for(int i = 1 ; i <=10 ; i ++){
			for(int j = 1 ; j <= i ; j ++){
				System.out.print(" "+1);
			}
		System.out.println(" ");
		}
	}
}
```

```
 1
 1 1
 1 1 1
 1 1 1 1
 1 1 1 1 1
 1 1 1 1 1 1
 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1
```

There are many types of operators in Java which are given below:

- Unary Operator,
- Arithmetic Operator,
- Shift Operator,
- Relational Operator,
- Bitwise Operator,
- Logical Operator,
- Ternary Operator and
- Assignment Operator.

There are many types of operators in JAVA

1.Unary Operator

2.Arithmetic Operator

3.Shift Operator

If **x=10**, then calculate **x>>2** value.

Shifting the value of x towards the right two positions will make the rightmost 2 bits to be lost. The value of x is 10. The binary representation of **10** is **00001010**. The procedure to do right shift explained in the following example:

Observe the above example, after shifting the bits to the right the binary number **00001010** (in decimal 10) becomes **00000010** (in decimal 2).

*or left side (**<< it adds the 0 at the left side**)*

TRIPLE RIGHT SHIFT

When we apply **>>>** on **a positive number**, it gives the same output as that of >>. It gives a positive number when we apply >>> on a negative number. MSB is replaced by a 0. Observe the above example, after shifting the bits to the right the binary number **00100000** (in decimal 32) becomes **00000100** (in decimal 4). The last three bits shifted out and lost.

4.Relational Operator

5.Bitwise Operator *(Is same as truth table)*
& *For e.g.: (10&4) --> 1010 & 0100 --> 0000 --> 0*
| *For e.g.: (10|4) --> 1010 | 0100 --> 1110 --> 12*
^ EXOR *for e.g. (10^7) --> 1010 ^ 0111 --> 1101 --> 13*

6.Logical Operator
&& if both condition is true then only it throughs true
|| if any of the condition is true then only it throughs true

7.Ternary Operator
?(if true then); :(if false then)
Condition1?print this : print this

8.Assignment Operator

----------SWITCH CASE----------

```
Switch(condition){
Case 1:
        SOP(Expression 1);
       break;
Case2:
        SOP(Expression 1);
       break;
Case3:
        SOP(Expression 1);
       break;
Default:
       SOP(Expression 1);
       break;
}
```

```
char a='A';

switch(a){
        case 'A' | 'a' | 'E' | 'e' | 'I' | 'i' | 'O' | 'o' | 'U' | 'u':
                System.out.println("Vowel");
        default:
                System.out.println("Not Vowel");
}
```

----------ARRAY----------

int a[] = new int[10]; //declaring an array
a[0] = 10; /* initializing the an array**/**
a[1] = 13;
a[2] = 15;
a[3] = 7;
System.out.print(a[0]);
System.out.print(a.length); //to print the length of an array.

```
coder@ubuntu:~/Desktop/Java Class$ java Hello
A [0] = 10
A [1] = 5
A [2] = 9
A [3] = 0
A [4] = 0
A [5] = 0
A [6] = 0
A [7] = 0
A [8] = 0
A [9] = 0
coder@ubuntu:~/Desktop/Java Class$
```

Import java.util.Scanner; *// to import scanner*
Scanner cin = new Scanner(System.in); *// we have to create object to use scanner from Scanner package system.in stands for system input stream which is again package of java present in java.lang*

```
import java.util.Scanner;

public class Hello{
        public static void main(String ar[]){
                Scanner cin = new Scanner(System.in);
                System.out.println("Enter a Number");
                int a = cin.nextInt();
                System.out.println("You Entered " + a);
        }
}
```

To read float Input.

```
import java.util.Scanner;

public class Hello{
        public static void main(String ar[]){
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter a float Number");
                float a = scan.nextFloat();
                System.out.println("You Entered " + a);
        }
}
```

To read String Input:

```
import java.util.Scanner;

public class Hello{
        public static void main(String ar[]){
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter a character");
                String a = scan.nextLine();
                System.out.println("You Entered " + a);
        }
}
```

Taking Input in array ;

```
import java.util.Scanner;

public class Hello{
        public static void main(String ar[]){
                Scanner scan = new Scanner(System.in);
                System.out.println("Enter size of array");
                int size=scan.nextInt();
                int array[] = new int[size];
                System.out.println("Enter "+ size +" element
                for (int i =0 ; i< size; i++){
                        array[i] = scan.nextInt();
                }
                for (int i =0 ; i< size; i++){
                        System.out.println("Array ["+i+"] =
                }
        }
}
```

```
I am Boy.
You Entered I am Boy.
coder@ubuntu:~/Desktop/Java
coder@ubuntu:~/Desktop/Java
Enter a Name
I am boy.
You Entered I
coder@ubuntu:~/Desktop/Java
Enter a Name
Amit Dash
You Entered Amit
coder@ubuntu:~/Desktop/Java
coder@ubuntu:~/Desktop/Java
Enter size of array
6
Enter 6 elements of Array
5 9 10 55 77 46
Array [0] = 5
Array [1] = 9
Array [2] = 10
Array [3] = 55
Array [4] = 77
Array [5] = 46
coder@ubuntu:~/Desktop/Java
```

----------2-D ARRAY----------

```
import java.util.Scanner;

public class Hello{
	public static void main(String ar[]){
		Scanner scan = new Scanner(System.in);
		int a[][] = new int[3][3];
		int b[][] = {{1,2,3}, {4,5,6}, {7,8,9}};
		for(int i =0; i<3; i++){
			for(int j = 0;j<3;j++){
				a[i][j]=scan.nextInt();
			}
		}

		for(int i =0; i<3; i++){
			for(int j = 0;j<3;j++){
				System.out.print(a[i][j] + " ");
			}
			System.out.println();
		}
		for(int i =0; i<3; i++){
			for(int j = 0;j<3;j++){
				System.out.print(b[i][j] + " ");
			}
			System.out.println();
		}
	}
}

/*
a00  a01  a02
a10  a11  a12
a20  a21  a22
```

OUTPUT:

```
coder@ubuntu:~/Desktop/Java Class$ java Hello
9 8 7 6 5 4 3 2 1
9 8 7
6 5 4
3 2 1
------------------------
1 2 3
4 5 6
7 8 9
```

int a[ ][ ]

int b[ ][ ]

Below is called as **TYPE CASTING**, basically assigning value to sum/size in our case.

```
System.out.println("Avg = "+(float)(sum/size));
```

----------FUNCTIONS----------

Creating a function called sum.

```
import java.util.Scanner;

public class Hello{

	public void sum (){
		int a=10,b=20,c;
		c=a+b;
		System.out.println("Sum = "+c);
	}
	public static void main(String ar[]){
		Hello h = new Hello();
		h.sum();
	}
}
```

```
coder@ubuntu:~/Desktop/Java Class$ java Hello
Sum = 30
```

```java
import java.util.Scanner;

public class Hello{
        int a,b;
        public void sum (){
                System.out.println(a+" "+b);
        }
        public void sum(int a, int b){
                System.out.println(a+b);
        }
        public static void main(String ar[]){
                Hello h = new Hello();
                h.sum();
                Scanner s = new Scanner(System.in);
                int n1 = s.nextInt();
                int n2 = s.nextInt();
                h.sum(n1,n2);
        }
}
```

Four different type of function:

```java
import java.util.Scanner;

public class Hello{

        public void sum1(){
                int a=20,b=10;
                System.out.println(a+b);
        }
        public int sum2(){
                int a=20,b=10;
                return (a+b);
        }
        public void sum3(int a, int b){
                System.out.println(a+b);
        }

        public int sum4(int a, int b){
                return (a+b);
        }
        public static void main(String ar[]){
                Hello h = new Hello();
                Scanner s = new Scanner(System.in);
                int n1 = s.nextInt();
                int n2 = s.nextInt();
                h.sum1();
                h.sum3(n1,n2);
                System.out.println(h.sum2());
                System.out.println(h.sum4(2,3));

        }
}
```

```
coder@ubuntu: ~/Desktop/Java Cl
5 6
11
coder@ubuntu:~/Desktop/Java Class$ javac Hello.java
Hello.java:12: error: method sum() is already defined
        public int sum(){
                   ^
Hello.java:16: error: method sum(int,int) is already
        public int sum(int a, int b){
                   ^
Hello.java:26: error: 'void' type not allowed here
                System.out.println(h.sum());
                                       ^
Hello.java:27: error: 'void' type not allowed here
                System.out.println(h.sum(2,3));
                                       ^
4 errors
coder@ubuntu:~/Desktop/Java Class$ javac Hello.java
coder@ubuntu:~/Desktop/Java Class$ java Hello
2 6
30
8
30
5
coder@ubuntu:~/Desktop/Java Class$
```

OVERRIDING (*In below case we have 4 function every function has same name but different type of parameter and number of parameter.)*

```java
import java.util.Scanner;

public class Hello{

        public void sum(int a, int b){
                System.out.println(a+b);
        }
        public void sum(int a, int b, int c){
                System.out.println(a+b+c);
        }
        public void sum(float a, float b){
                System.out.println(a+b);
        }
        public void sum(float a, float b, float c){
                System.out.println(a+b+c);
        }
        public void sum(int a, float b){
                System.out.println(a+b);
        }
        public void sum(float a, int b){
                System.out.println(a+b);
        }
        public static void main(String ar[]){
                Hello h = new Hello();
/*              Scanner s = new Scanner(System.in);
                int n1 = s.nextInt();
                int n2 = s.nextInt();*/
                h.sum(1,2);
                h.sum(1,2,3);
                h.sum(1.0f,2.0f);
                h.sum(1.0f,2.0f,3.0f);
                h.sum(1,2.0f);
```

OUTPUT:

```
coder@ubuntu:~/Desktop/Java Class$ java Hello
3
6
3.0
6.0
3.0
3.0
```

You can also use this syntax to call a function in main without creating an object name aka instance.

```java
new Hello().sum(1,2);
```