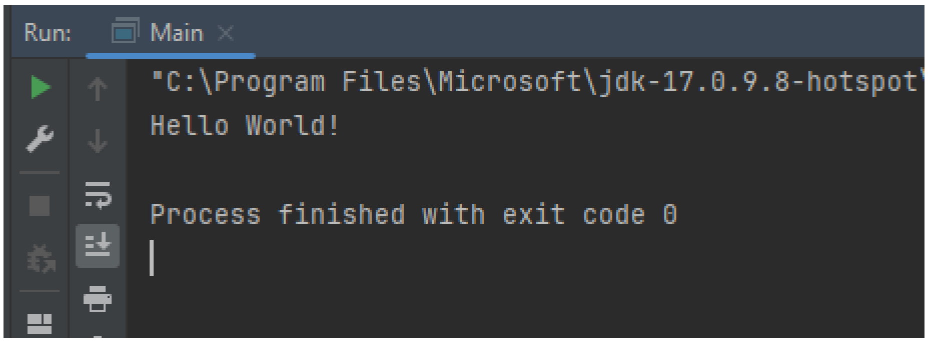
Java Full Stack

Hello World Program without maven

Code:

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

       System.out.println("Hello World!");

       }
   }
```



```
"C:\Program Files\Microsoft\jdk-17.0.9.8-hotspot
Hello World!

Process finished with exit code 0
```

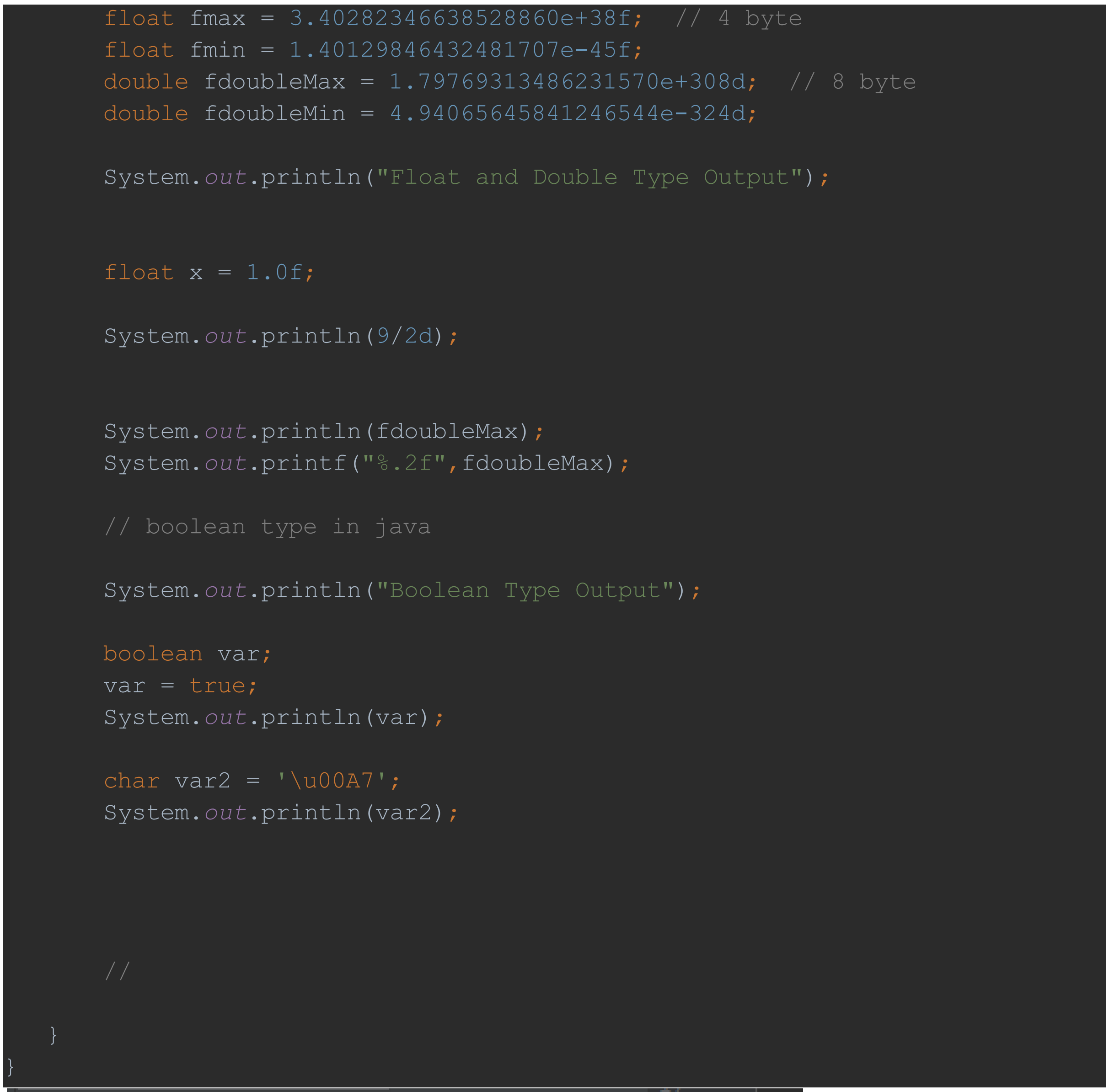
Keywords in java: some of the famous keywords are familiar.

Understanding variables in java.

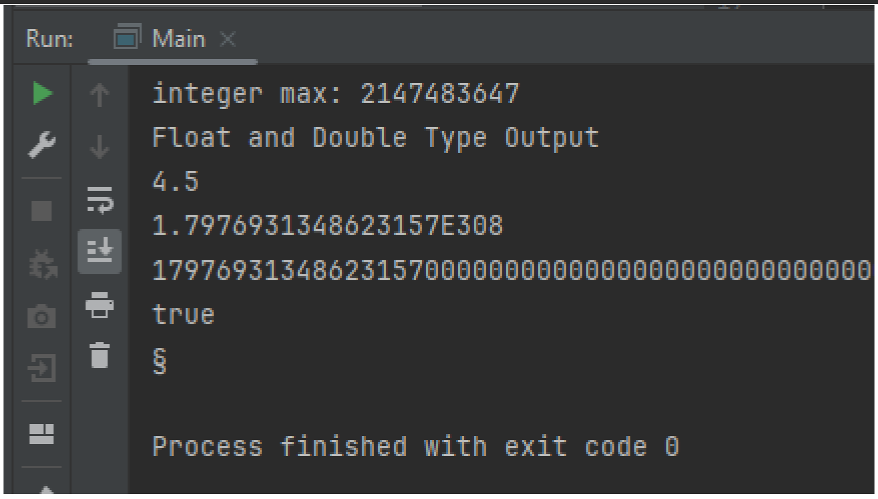
Refer source for Understanding variables.

Data Types Code

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

       int max = 2147483647; // 4 byte
       int min = -2147483648;
       short shortMax = 32767; //2 byte
       short shortMin = -32768;
       long longMax = 9223372036854775807L; // 8 byte
       long longMin = -9223372036854775808L;
       byte byteMax = 127; // 1 byte
       byte byteMin = -128;

       System.out.println("Int Type Output");


       System.out.println("integer max: "+max);

       // float data type
```



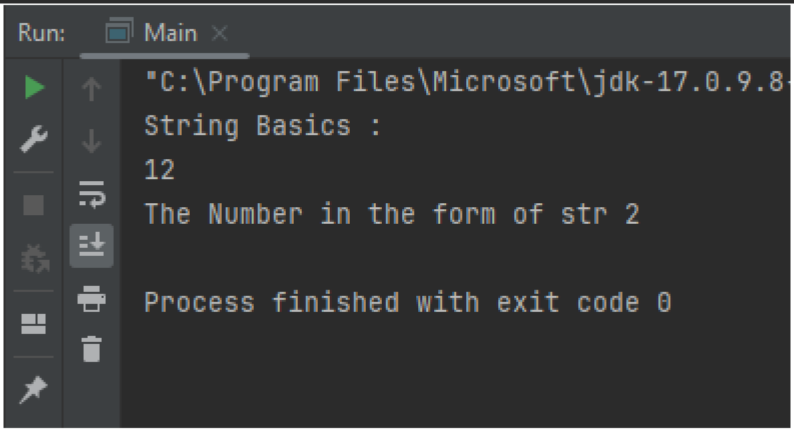
```
        float fmax = 3.40282346638528860e+38f;  // 4 byte
        float fmin = 1.40129846432481707e-45f;
        double fdoubleMax = 1.79769313486231570e+308d;  // 8 byte
        double fdoubleMin = 4.94065645841246544e-324d;

        System.out.println("Float and Double Type Output");


        float x = 1.0f;

        System.out.println(9/2d);


        System.out.println(fdoubleMax);
        System.out.printf("%.2f",fdoubleMax);

        // boolean type in java

        System.out.println("Boolean Type Output");

        boolean var;
        var = true;
        System.out.println(var);

        char var2 = '\u00A7';
        System.out.println(var2);




        //

    }
}
```



```
integer max: 2147483647
Float and Double Type Output
4.5
1.7976931348623157E308
179769313486231570000000000000000000000000000000
true
§

Process finished with exit code 0
```

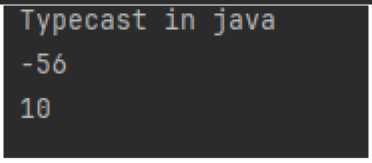
String Basics:

```java
package org.example;


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

       System.out.println("String Basics : ");

       String name = "Mahesh \u00f1";
       String no = "2";
       int n = 20;
       System.out.println(Integer.parseInt(no)+10);
       System.out.println("The Number in the form of str "+no);


   }
}
```



```
"C:\Program Files\Microsoft\jdk-17.0.9.8
String Basics :
12
The Number in the form of str 2

Process finished with exit code 0
```

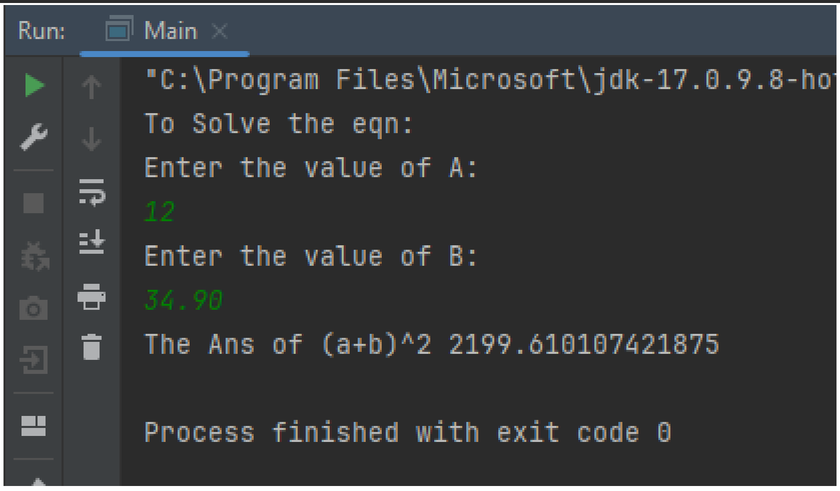
```java
// typecasting in java from string to int and all

System.out.println("Typecast in java ");

short a1 = 200;
byte a2 = (byte)a1;
System.out.println(a2);

// typecast 2

float b = 10.5f;
int i = (int)b;
System.out.println(i);
```



```
Typecast in java
-56
10
```

To Solve the eqn

```
package org.example;
import java.util.*;
public class Main {
   public static void main(String[] args) {

       // to solve an equation
       System.out.println("To Solve the eqn:");

       //(a+b)^2=a^2+b^2+2ab;

       Scanner sc = new Scanner(System.in);
       System.out.println("Enter the value of A: ");
       int a = sc.nextInt();

       System.out.println("Enter the value of B: ");
       float b= sc.nextFloat();

       double val = a*a + 2*a*b + b*b;

       System.out.println("The Ans of (a+b)^2 "+val);




   }
}
```

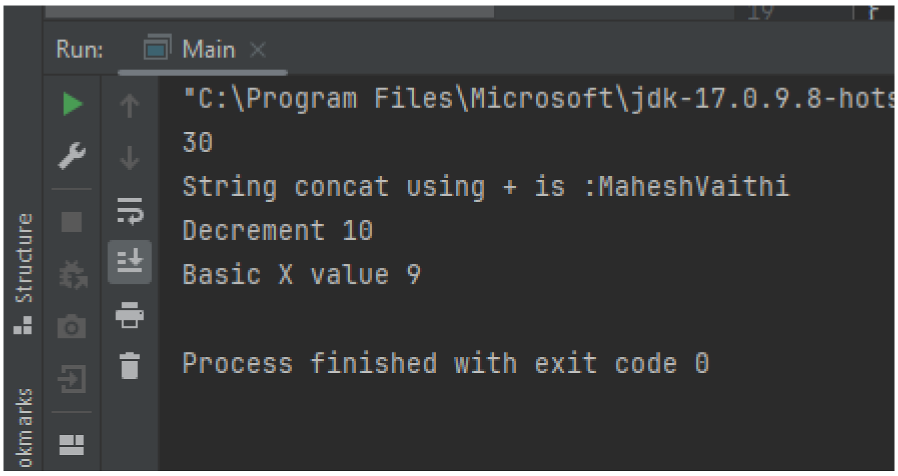


```
"C:\Program Files\Microsoft\jdk-17.0.9.8-ho
To Solve the eqn:
Enter the value of A:
12
Enter the value of B:
34.90
The Ans of (a+b)^2 2199.610107421875

Process finished with exit code 0
```

Arithmetic Operators in Java:

```
package org.example;
public class Main {
   public static void main(String[] args) {

       // arithmetic operator

       int x = 300/10;
```

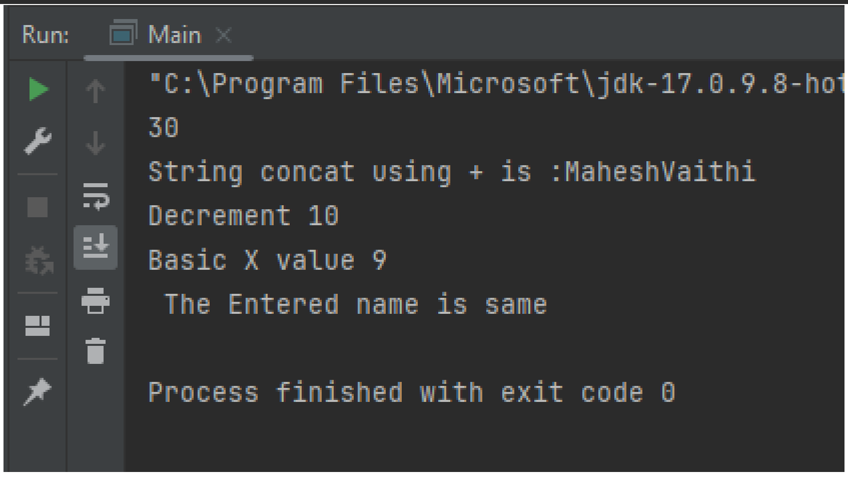
```
        String text = "Mahesh" + "Vaithi";
        System.out.println(x);

        // 2
        int mod = 16%4;
        int a = 10;
        System.out.println("Decrement  "+a--);
        System.out.println("Basic X value "+a);

    }
}
```



```
"C:\Program Files\Microsoft\jdk-17.0.9.8-hots
30
String concat using + is :MaheshVaithi
Decrement 10
Basic X value 9

Process finished with exit code 0
```

```
package org.example;
public class Main {
   public static void main(String[] args) {

        // arithmetic operator

        int x = 300/10;
        String text = "Mahesh" + "Vaithi";
        System.out.println(x);
        System.out.println("String concat using + is :"+text);

        // 2
        int mod = 16%4;
        int a = 10;
        System.out.println("Decrement  "+a--);
        System.out.println("Basic X value "+a);

        // ternary operator


        String st = "Mahesh";

       String res = (st=="Mahesh") ? "The Entered name is same" : "The name is
Diff";

        System.out.println(" "+res);
```

```
    }
}
```

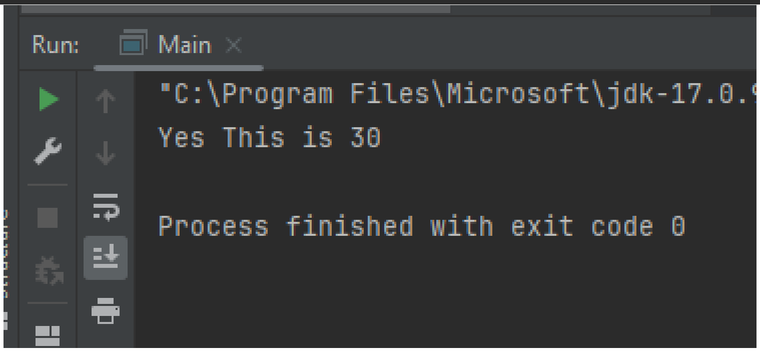


```
"C:\Program Files\Microsoft\jdk-17.0.9.8-hot
30
String concat using + is :MaheshVaithi
Decrement 10
Basic X value 9
 The Entered name is same

Process finished with exit code 0
```

Basic Control Statements are familiar with loops and control statements

Switch statement with lambda exp;

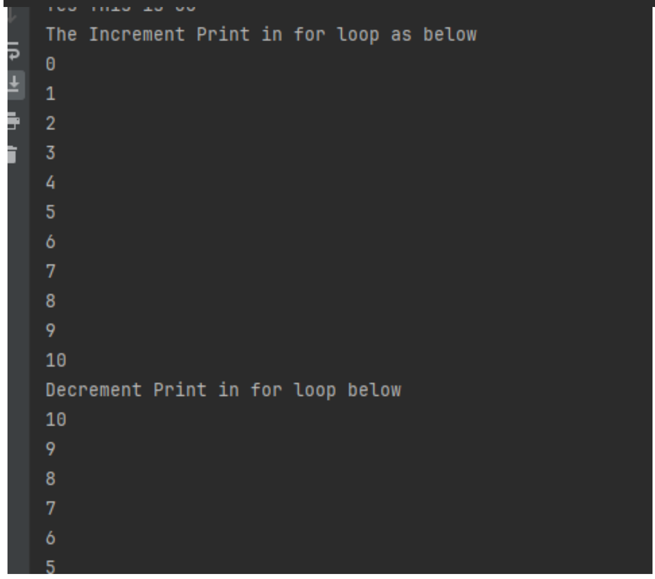
```java
package org.example;

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

        // switch statements in java
        // lambda exp

        int x=30;

        switch(x) {

            case 1 -> System.out.println("This is not 30");
            case 30 -> System.out.println("Yes This is 30");
            default -> System.out.println("None");
        }

    }
}
```



```
"C:\Program Files\Microsoft\jdk-17.0.
Yes This is 30

Process finished with exit code 0
```

Loops are familiar

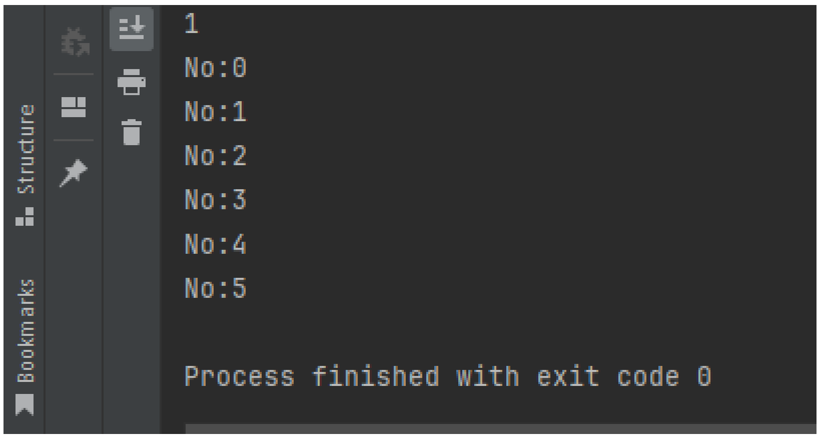
```
// for loop in java

// increment print

System.out.println("The Increment Print in for loop as below");
for(int j=0;j<=10;j++){
   System.out.println(j);
}

// decrement print

System.out.println("Decrement Print in for loop below");

for(int s=10;s>=1;s--){
   System.out.println(s);
}
```



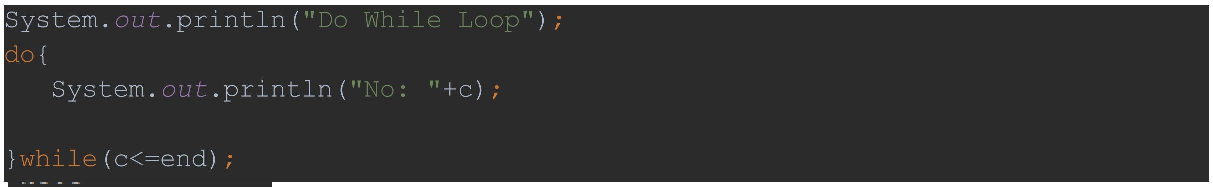
```
The Increment Print in for loop as below
0
1
2
3
4
5
6
7
8
9
10
Decrement Print in for loop below
10
9
8
7
6
5
```

While and do while loop are also familiar

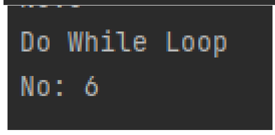
```
// while loop and do while loop

int end=5;
int c=0;

while(c<=end){
   System.out.println("No:"+c);
   c=c+1;
}
```



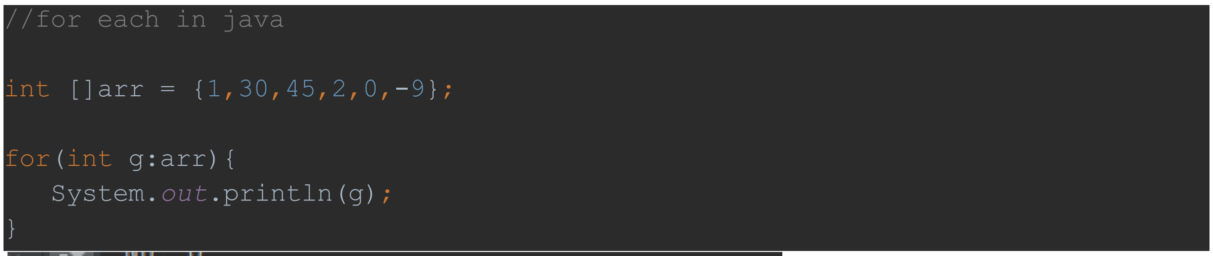
```
1
No:0
No:1
No:2
No:3
No:4
No:5

Process finished with exit code 0
```



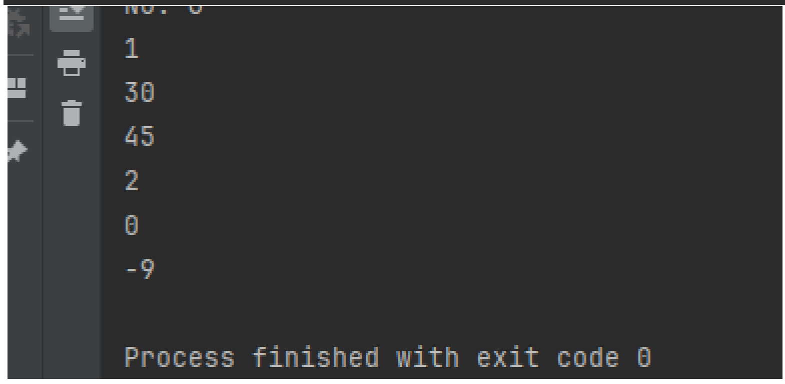
```
System.out.println("Do While Loop");
do{
   System.out.println("No: "+c);

}while(c<=end);
```



```
Do While Loop
No: 6
```



```
//for each in java

int []arr = {1,30,45,2,0,-9};

for(int g:arr){
   System.out.println(g);
}
```



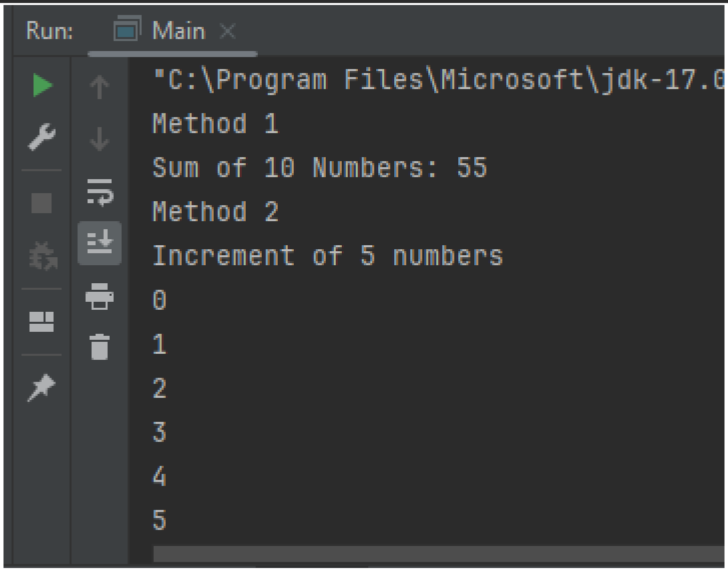
```
1
30
45
2
0
-9

Process finished with exit code 0
```

Nested Loops are already familiar ;

Java Methods :

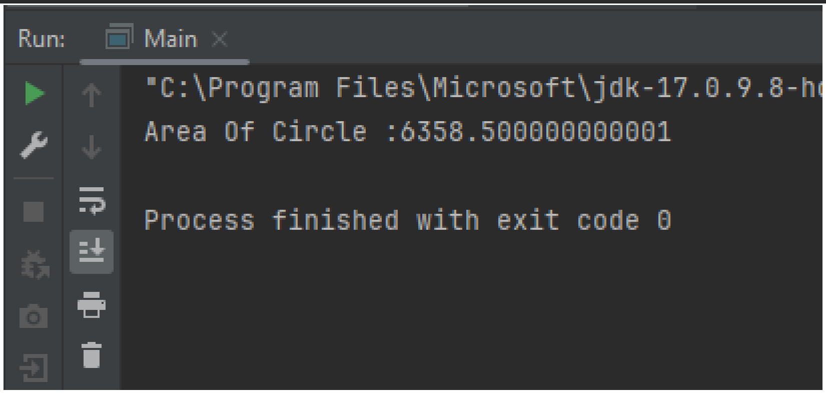
Method Overloading
Overriding , Are some basic concepts:

```java
package org.example;
public class Main {
   public static void main(String[] args) {
       // Java Methods

       loopsum();

       loop();


   }

   public static void loopsum(){
       int s=0;
       System.out.println("Method 1");
       for(int i=0;i<=10;i++){
           s=s+i;
       }

       System.out.println("Sum of 10 Numbers: "+s);
   }

   public static void loop(){
       System.out.println("Method 2");
       System.out.println("Increment of 5 numbers");
       for(int j=0;j<=5;j++){
           System.out.println(j);
       }
   }
}
```



```
"C:\Program Files\Microsoft\jdk-17.0
Method 1
Sum of 10 Numbers: 55
Method 2
Increment of 5 numbers
0
1
2
3
4
5
```

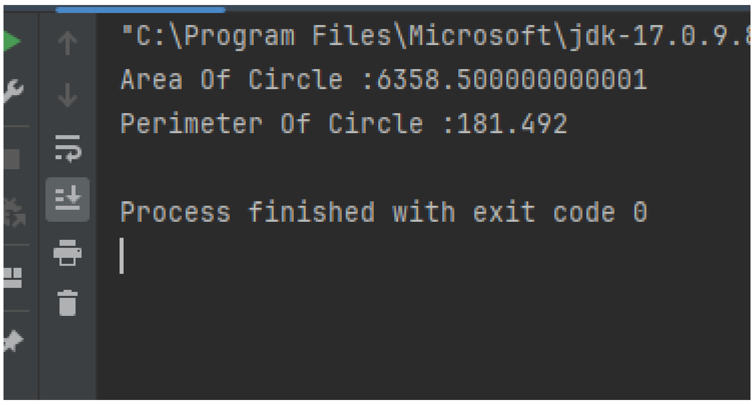
```java
// return value in method

public static double area(int radius) {
   return 3.14*radius*radius;
}
public static void main(String[] args) {
   // Java Methods

   //loopsum();

   //loop();

   System.out.println("Area Of Circle :"+area(45));


}
```



```
"C:\Program Files\Microsoft\jdk-17.0.9.8-h
Area Of Circle :6358.500000000001

Process finished with exit code 0
```

Method Overloading:

```
// method overloading

public static double circle(int radius) {
   return 3.14*radius*radius;
}



public static double circle(double r){
   return 2*3.14*r;

}
```



```
"C:\Program Files\Microsoft\jdk-17.0.9.
Area Of Circle :6358.500000000001
Perimeter Of Circle :181.492

Process finished with exit code 0
```

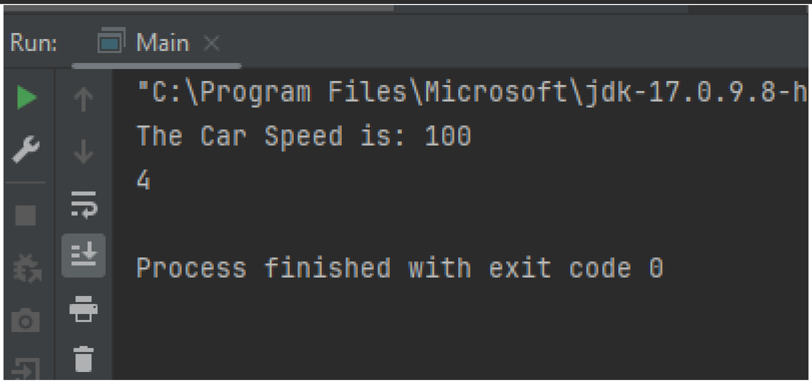
Method overloaded with same method name with different params.

Java OOPS Concept:

Classes and Objects:

```
package org.example;

public class Main {
   public static void main(String[] args) {
       // class initialization
       Car c = new Car();
       //c.getdoors(4);
       c.speed = 100;
       System.out.println("The Car Speed is: "+c.speed);

       //getter and setter in java

       System.out.println(c.getdoors(4));



   }
}
```

```
package org.example;

public class Car {

   private int doors;
   private String driver;
   public int speed;
   // get and set
   public int getdoors(int doors){
       return doors;
   }
}
```



```
"C:\Program Files\Microsoft\jdk-17.0.9.8-h
The Car Speed is: 100
4

Process finished with exit code 0
```

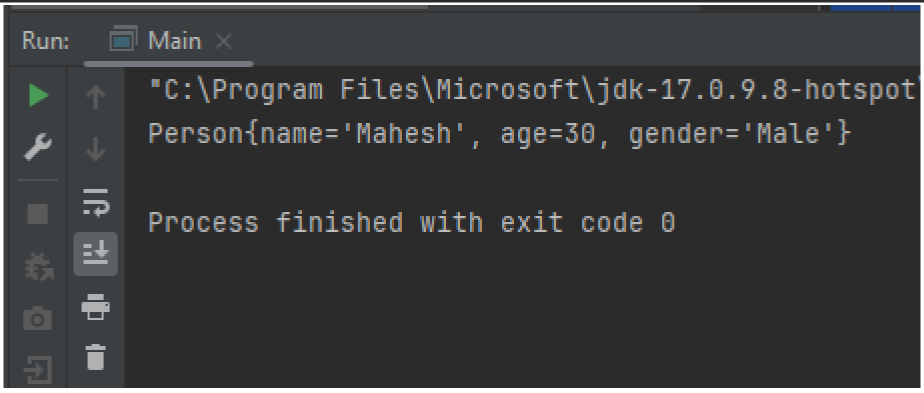
For Inheritance refer the code:

Composition in Java:

Codes are done on composite concepts:

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

       Person p = new Person("Mahesh",25,"Male");

       p.setAge(30);
       System.out.println(p);
   }
}
package org.example;

public class Person {
   private String name;
   private int age;



   private String gender;

   public Person(String name, int age, String gender) {
       this.name = name;
       this.age = age;
       this.gender = gender;
   }

   public boolean setAge(int age) {
       if(age>0 && age<=100){

           this.age = age;
           return true;
```

```
        }
        else {
            return false;
        }

    }

    @Override
    public String toString() {
        return "Person{" +
                "name='" + name + '\'' +
                ", age=" + age +
                ", gender='" + gender + '\'' +
                '}';
    }
}
```

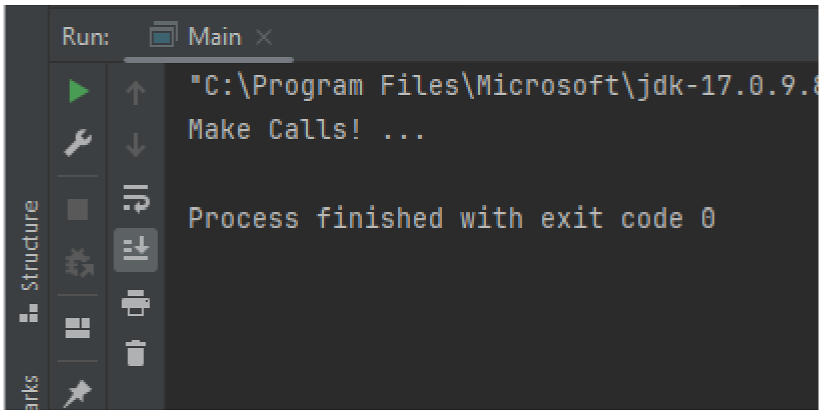


```
"C:\Program Files\Microsoft\jdk-17.0.9.8-hotspot
Person{name='Mahesh', age=30, gender='Male'}

Process finished with exit code 0
```

Polymorphism

```
package org.example;

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

        //polymorphism

        /*Phone p = new Phone();
        p.feature();

        PocoMobile pc = new PocoMobile();
        pc.feature();

        MotoMobile m = new MotoMobile();
        m.feature();*/
        Phone p;
        p = new Phone();
        p.feature();
```

```java
    }
}
package org.example;

public class Phone {
    public void feature(){
        System.out.println("Make Calls! ...");
    }
}
package org.example;

public class MotoMobile extends Phone{
    public void feature()
    {
        System.out.println("Make Calls more Reliable...");
    }
}
package org.example;

public class PocoMobile extends Phone {
    public void feature()
    {
        System.out.println("Make Calls and do Smart Things...");
    }
}
```



```
"C:\Program Files\Microsoft\jdk-17.0.9.
Make Calls! ...

Process finished with exit code 0
```

Interface with multiple inheritance examples are done.