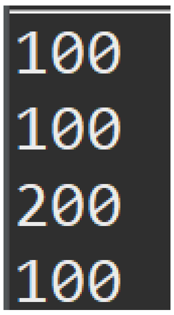
Value type assignment

```
class Test2
{
	public static void main(String[] args)
	{
			int x = 100; // Assigning the value to the variable x
			int y; // Declaring the variable y

			y = x;  // assigning the value present in x to y

			System.out.println(x); // print the value of x
			System.out.println(y); // print the value of y

			x = 200;  // modifying the value of x

			System.out.println(x); // print the value of x
			System.out.println(y); // print the value of y
	}
}
```

Output:



```
100
100
200
100
```

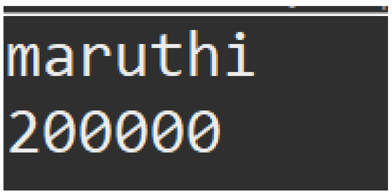
If you observe from the above output, if you change the value present in one variable it will not affect other variable

Reference type assignmemt

- **Let’s create two instance variable name and cost and assign values to those variables**

```
class Car
{
	String name; // instance variable name is created of type string
	int cost; // instance variable cost is created of type int
}

class Test2
{
	public static void main(String[] args)
	{
		Car x = new Car(); // car class object is created
		x.name = "maruthi"; // assigning value to the instance variable name using object reference
		x.cost = 200000; // assigning value to the instance variable cost using object reference

		System.out.println(x.name); // print the value present in instance variable
		System.out.println(x.cost); // print the value present in instance variable

	}
}
```

Output:

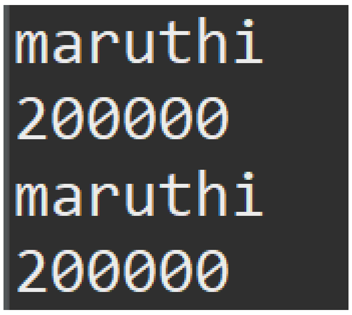


```
maruthi
200000
```

- Let’s create one more object reference y of type car and assign the reference present in x to y.

```
class Car
{
        String name; // instance variable name is created of type string
        int cost;       // instance variable cost is created of type int
}

class Test2
{
        public static void main(String[] args)
        {
                Car x = new Car(); // car class object is created
                x.name = "maruthi"; // assigning value to the instance variable
name using object reference
                x.cost = 200000; // assigning value to the instance variable cost
using object reference

                System.out.println(x.name); // print the value present in instance
variable name
                System.out.println(x.cost); // print the value present in instance
variable cost

                Car y;  // create an object reference of type y
                y = x;  // assign the reference present inside x to y

                System.out.println(y.name); // print the value present in instance
variable name using object reference y
                System.out.println(y.cost); // print the value present in instance
variable cost using object reference y

        }
}
```

Output:



```
maruthi
200000
maruthi
200000
```

If you observe from the above output when you print the values using object reference x and y it is giving you same result it is because both are pointing to same object

- **Now both the object reference x and y are pointing to the same object. Let’s try to modify the value of instance variable using one reference**

```
class Car
{
        String name;
        int cost;
}

class Test2
{
        public static void main(String[] args)
        {
                Car x = new Car();
                x.name = "maruthi";
                x.cost = 200000;

                System.out.println(x.name);
                System.out.println(x.cost);

                Car y;
                y = x;

                System.out.println(y.name);
                System.out.println(y.cost);

                y.name = "BMW"; // now let's modify the instance variable name
using object reference y
                y.cost = 500000; // now let's modify the instance variable cost
using object reference y

                System.out.println(x.name); // print the value of name using
object reference x
                System.out.println(x.cost); // print the value of cost using object
reference x
        }
}
```

Output:

```
maruthi
200000
maruthi
200000
BMW
500000
```

If you observe from the above output if you modified the value using one object reference it will effect all other references pointing to the same object

Pass by value

- **Create a method which accepts two parameters and return the result**

```
class Calculator
{
     int c; // create an instance variable c of type int

          // define the method add which accepts two parameters perform addition operation and return the result
     int add(int a, int b)
     {
          c = a + b;
          return c;
     }

     public static void main(String[] args)
     {
          Calculator calc = new Calculator(); // create an object of class calculator
          int num1 = 50; // assign value 50 to the variable num1
          int num2 = 40; //assign the value 40 to the variable num2
          int res = calc.add(num1, num2); // call add method and store the returned value store it in res
          System.out.println(res); //print res

          }
}
```

Output:

```
90
```

```
class Calculator
{
     int c;
     int add(int a, int b)
     {
          a = 500; // here local variable value is 
assigned with value 500
          b = 400; // here local variable value is 
assigned with value 400
          c = a + b; //Addition of two variable a and b 
is done and store the result in c
          return c; // result the value present in c
     }

     public static void main(String[] args)
     {
          Calculator calc = new Calculator(); // create 
an object of class
          int num1 = 50; // declare local variable num1 
and assign the value 10
          int num2 = 40; // declare local variable num1 
and assign the value 10
          int res = calc.add(num1, num2); //call the 
method by passing the value present in num1 and num2
          System.out.println(res); //print res that is 
returned by the method

          }
}
```

Output:

```
900
```

- **Now let’s try to create the method which accept the reference of type car and modify the value of instance variable inside that method**

```
class Car
{

	String name;
	int cost;

	void modifyCar(Car y)
	{
		y.name = "BMW";
		y.cost = 500000;
	}
}

class Test2
{
	public static void main(String[] args)
	{
		Car x = new Car();
		x.name = "maruthi";
		x.cost = 200000;

		System.out.println(x.name);
		System.out.println(x.cost);

		x.modifyCar(x);

		System.out.println(x.name);
		System.out.println(x.cost);

	}
}
```

Output:

```
maruthi
200000
BMW
500000
```

Here if you observe from the above output the you have updated the value by using the reference y now if you try to access it using the reference x you are getting updated value it is because of pass by reference