1. Arguments created in methods are treated as local variables

2. var datatype cannot be method argument

3. What happens if the datatype of the variable is made Object ?

Ans: Then the variable can store any value, just like var datatype

Example 1:

```
public class A {

	public static void main(String[] args) {
		A a1 = new A();
		a1.test(100);
	}

	public void test(int x){
		System.out.println(x);
	}

}
```

Output:

100

Example 2:

```
public class A {

	public static void main(String[] args) {
		A a1 = new A();
```

```
		a1.test(100,200, "Pankaj");
	}

	public void test(int x, int y, String z){
		System.out.println(x);
		System.out.println(y);
		System.out.println(z);
	}

}
```

Output:

100

200

Pankaj

Example 3:

```
public class A {

	public static void main(String[] args) {
		A a1 = new A();
		a1.test(100,200,300,400);
	}
	public void test(int... x){
		System.out.println(x[0]);
		System.out.println(x[1]);
		System.out.println(x[2]);
```

```
            System.out.println(x[3]);
        }
}
```

Output:

100

200

300

400

Example 4:

```
public class A {

        public static void main(String[] args) {
            A a1 = new A();
            a1.test(100);
        }
        public void test(var x){
            System.out.println(x);

        }
}
```

Output: Error

Example 5:

```
public class A {
	public static void main(String[] args) {
		Object x = 10;
		System.out.println(x);
		Object y = 10.3;
		System.out.println(y);
		Object z = "String";
		System.out.println(z);
		Object t = true;
		System.out.println(t);
		Object f = 'a';
		System.out.println(f);
	}
}
```

Output:

10

10.3

String

true

a

Example 6:

```
public class A {
```

```
	public static void main(String[] args) {
		A a1 = new A();
		a1.test(100, "PankajSir Academy",10.3,true);
	}

	public void test(Object... x){
		System.out.println(x[0]);
		System.out.println(x[1]);
		System.out.println(x[2]);
		System.out.println(x[3]);
	}
}
```

Output:

100

PankajSir Academy

10.3

true

Constructors:

1. Constructors should have same name as that of class.

2.