```java
package com.instancepractice;
class Employee{
	static int id=1000;
	 static void display()
	{
		System.out.println("ID:"+id);
	}
}
public class StaticMethod {

	public static void main(String[] args) {
		Employee.id=100;
		Employee.display();
	}

}
```

```java
package com.instancepractice;
class Test{
	int x=900;
	void add(){
		System.out.println("result :"+x);
	}
}
public class AnonymousObject {

	public static void main(String[] args) {
		Test t1=new Test();
		t1.x=500;
		t1.add();
		t1.add();
		t1.add();
		 System.out.println("-----");

		 new Test().x=400;
		 new Test().add();
		 new Test().add();
		 new Test().add();

	}

}
```

```java
package com.instancepractice;

public class ConstructorPractice1 {
	int id;
	String name;
```

```
	/*
	 * ConstructorPractice1(int i,String n) { id = i; name = n;
	 *
	 * }
	 */

	/*
	 * ConstructorPractice1(ConstructorPractice1 c) { id=c.id;
name=c.name; }
	 */
	/*
	 * ConstructorPractice1() {
	 *
	 * }
	 */
	void display()
	{
		System.out.println(id+" "+name);
	}
	public static void main(String[] args) {
		ConstructorPractice1 c1=new ConstructorPractice1();
		ConstructorPractice1 c2=new ConstructorPractice1();
		c2.id=c1.id;
		c2.name=c1.name;
		c1.display();
		c2.display();
	}

}



package com.instancepractice;
class Student{
	/*static*/ int id;
	/* static */  void display()
	{
		System.out.println("Result:"+id);
	}

}
public class ObjCreation {

	public static void main(String[] args) {
		/*Student.id=300;				//Calling a static
method,static variable
		Student.display();
		System.out.println("------");*/
```

```
            Student s1=new Student();   //Creating object for the
class Student
            s1.id=100;
            s1.display();
            System.out.println("-------");

            Student s2=new Student();   //Creating s2 obj for same
student class
            s2.id=200;
            s2.display();

      }

}
```

```
package com.instancepractice;       //upcasting & downcasting
class A
{
      int x;
      void sum() {
            System.out.println("A: x="+x);
      }
}
class B extends A{
      int y;
      void add()
      {
            System.out.println("B: X="+x+"  Y="+y);
      }
}
public class ObjectReference1 {

      public static void main(String[] args) {
                  B s1=new B();

                  s1.x=100;
                  s1.sum();

                  s1.y=459;
                  s1.add();
                  System.out.println("-------");

                  A s2=new B();  //upcasting  (child parameters
used by parent)
                  s2.x=50;
                  s2.sum();
                  System.out.println("-------");
```

```
                    // A s=new A();
                     B s3=(B)s2;
                     s3.sum();   //Downcasting-parent parameters used
by child
                                 //Updating must be done before
downcasting
                     System.out.println("-------");

                     s3.y=400;
                     s3.add();


      }

}


package com.instancepractice;
class C{
      int x;
      void add() {
            System.out.println("C: x="+x);
      }
}
class D extends C{
      int y;
      void sum() {
            System.out.println("D: X="+x+" Y:"+y);
      }
}
public class ObjectReference2 {

      public static void main(String[] args) {
                  D r1=new D();
                  r1.x=100;
                  r1.add();

                  r1.y=200;
                  r1.sum();
                  System.out.println("-------");

                  C r2=r1;
                  r2.add();
                  //r2.sum();
                  System.out.println("-------");

                  D r3=(D)r2;
                  r3.add();
                  r3.sum();
```

```
	}

}
```

```
package com.instancepractice;
class Ddemo{
	int x=10;
	void add()
	{
		System.out.println(x);
	}
	void add(int x)
	{
		this.x=x;
		System.out.println("X="+x);
	}
}
public class PracticeMethodOverload1 {

	public static void main(String[] args) {
	   Ddemo ddo=new Ddemo();
	   ddo.add();
	   ddo.add(50);
	}

}
```

```
package com.instancepractice;
class Aa{
	void add() {
	System.out.println("a");
	}
}
class Bb extends Aa
{
	void add()
	{
		System.out.println("b");
	}
}
public class PracticeMethodOverride {

	public static void main(String[] args) {
		Bb o1=new Bb();
	    o1.add();
	   Aa o2=new Aa();
	    o2.add();
	   Aa o3=new Bb();
```

```
        o3.add();
        Aa o4=o1;
        o4.add();
        Aa o5=(Aa)o1;
        o5.add();
        Aa o6=(Aa)new Bb();
        o6.add();
      //  Bb o7=o2;   //error
        Bb o8=(Bb)o4;    //don't put Bb o8=(Bb)o2;
        o8.add();
      }

}
```

```
package com.instancepractice;

public class PracticeTypeCasting1 {

      public static void main(String[] args) {
            int a=100;
            long l1=a;
            long l2=(long)l1;
            System.out.println(l1);
            System.out.println(l2);

            char b=(char)a;
            System.out.println(b);

            double d=100.45;
            long l=(long)d;
            float f=(float)d;
            int g=(int)l;
            System.out.println(d+" "+l+" "+f+" "+g);
      }

}
```

```
package com.instancepractice;

public class TypeCasting {

      public static void main(String[] args) {
             int a=100;
             char b=(char)a;
             System.out.println("Integer to Char:"+b+"\n");

             double d=100.04;
             int i=(int)d;     //explicit type casting required
             long l=(long)i;   //explicit type casting required
```

```
            System.out.println("Double value:"+d);
            System.out.println("Integer value:"+i);
            System.out.println("Long value:"+l);

            System.out.println();

            int i2 = 100;
            long l2 = i2;  //no explicit type casting required
            float f2 =l2;  //no explicit type casting required
            double d2=f2;

            System.out.println("Int value "+i2);
            System.out.println("Long value "+l2);
            System.out.println("Float value "+f2);
            System.out.println("Double value "+d2);
        }

}
```