Type Casting

Process of converting one type of information into another type is known as type-casting. There are of two types:
1. Data type casting
2. Class type casting

Data Type casting

The process of converting one data type into another data type is known as data type casting.
There are of two types:
1. Narrowing
2. Widening

Narrowing:

Process of converting higher level datatype into lower level data type is known as narrowing.

Widening:

Process of converting lower level datatype into higher level data type is known as widening.

The process of converting one type of information into another type by compiler is known as implicit type conversion.

The process of converting one type of information into another type by programmer is known as explicit type conversion.

Widening can be done both explicitly and implicitly. Narrowing must be done explicitly.

Class Type casting

Process of converting one class type into another class type is known as class type casting.
There are of two types:
1. Upcasting
2. Downcasting

Upcasting

Process of converting subclass type into superclass type is known as upcasting. Upcasting can be done both explicitly and implicitly.
After upcasting operation we can access only superclass properties.

Downcasting

Process of converting superclass type into subclass type is known as Downcasting. Downcasting must be done explicitly.
After downcasting operation we can access both subclass and superclass properties.

IS-A relationship is mandatory to achieve class type casting.

Example for widening:

```
Int x;
Double y=(double)x;
```

Example for narrowing:

Double x;
Int y=(int)x;

1. Program for widening and narrowing

```
public class TypeCastingMain2
{
public static void main(String args[])
{
// widening
        int x=10;
        double y=(double)x;
        System.out.println("The widening of x is"+y);
// Narrowing
        double a=2.4;
        int b=(int)a;
        System.out.println("The narrowing of a is"+b);
}
}
```

2. Type casting while passing arguments

```
public class TypeCastingMain3
{
        public static void test(int x)
        {
                System.out.println("The value of x is"+x);
        }
        public static void main(String[] args)
        {
                double d=2.8;
                test((int)d);
        }
}
```

3. Type casting return type

```
public class TypeCastingMain4
{
public static int test()
{
        return(int)5.6;
}
public static void main(String args[])
{
        int x=test();
        System.out.println(x);
}
}
```

4. Type casting in System.out.println

```
public class TypeCastingMain5
{
	public static void main(String[] args)
	{
		for(int i=100;i<=110;i++)
			{
			System.out.println((char)i);
			System.out.println((double)i);
			}
	}

}
```

Upcasting

Eg for upcasting

```
public class Vehicle
{
	String Vehicle_Type;
	String Reg_no;
	String Insurace_Name;

	Vehicle(String a, String b, String c)
	{
		Vehicle_Type=a;
		Reg_no=b;
		Insurace_Name=c;
	}
}
public class Bike extends Vehicle
{
String Type;

Bike(String T)
{
	super("Bike","KA-01","New India");
	Type=T;
}
}
public class Car extends Vehicle
{
String Type;

Car(String T)
{
	super("Car","KA-09","Oriental");
	Type=T;
}
}
public class TypeCastingMain1
{
	public static void main(String args[])
	{
		//upcasting
```

```
		Vehicle v=(Vehicle) new Car("Car");
		Vehicle v1=(Vehicle) new Bike("Bike");

		System.out.println(v.Vehicle_Type +"\t"+v.Reg_no+"\t"+
v.Insurace_Name);

	System.out.println("=====================================");
		System.out.println(v1.Vehicle_Type +"\t"+v1.Reg_no+"\t"+
v1.Insurace_Name);
	}
}
```

Downcasting

Downcating process first we need to upcast an object, then we need to downcast, since super class wont contain all the properties of a subclass.

```
public class Person
{
String Name="Alex";
}
public class Student extends Person
{
double marks=65.5;
}
public class TypeCastingMain6
{
	public static void main(String[] args)
	{
		//upcasting
		Person p=(Person) new Student();
		//downcasting
		Student s=(Student)p;
		System.out.println("Student name: "+s.Name);
		System.out.println("Student Marks: "+s.marks);
	}
}
```

1. Write downcasting with application and gmail

```
public class Application
{
	String App_name;
	Application(String App_name)
	{
		this.App_name=App_name;
	}
}
public class Gmail extends Application
{
double rating;
double user;

Gmail(double rating, double user)
{
	super("Gmail");
	this.rating=rating;
	this.user=user;
```

```
}
}
public class TypeCastingMain7 {

	public static void main(String[] args)
	{
		Application a=(Application)new Gmail(4.5,12000);
		Gmail g=(Gmail) a;
		System.out.println("Application name"+g.App_name);
		System.out.println("Application rating"+g.rating);
		System.out.println("Application users"+g.user);

	}

}
```