**Usage of “super(args)” during implementation**

Below gives a perfect example on the usage of super. The programmer uses only the subclass to determine every declarations:

**Class 1 – DemoSuper.java:**

**class** DemoSuper {

**public** **static** **void** main(String args[]) {

BoxWeight mybox1 = **new** BoxWeight(10, 20, 15, 34.3);

BoxWeight mybox2 = **new** BoxWeight(2, 3, 4, 0.076);

BoxWeight mybox3 = **new** BoxWeight(); // default

BoxWeight mycube = **new** BoxWeight(3, 2);

BoxWeight myclone = **new** BoxWeight(mybox1);

**double** vol;

vol = mybox1.volume();

System.*out*.println("Volume of mybox1 is " + vol);

System.*out*.println("Weight of mybox1 is " + mybox1.weight);

System.*out*.println();

vol = mybox2.volume();

System.*out*.println("Volume of mybox2 is " + vol);

System.*out*.println("Weight of mybox2 is " + mybox2.weight);

System.*out*.println();

vol = mybox3.volume();

System.*out*.println("Volume of mybox3 is " + vol);

System.*out*.println("Weight of mybox3 is " + mybox3.weight);

System.*out*.println();

vol = myclone.volume();

System.*out*.println("Volume of myclone is " + vol);

System.*out*.println("Weight of myclone is " + myclone.weight);

System.*out*.println();

vol = mycube.volume();

System.*out*.println("Volume of mycube is " + vol);

System.*out*.println("Weight of mycube is " + mycube.weight);

System.*out*.println();

}

}

**Class 2 – Box.java:**

**public** **class** Box {

**private** **double** width;

**private** **double** height;

**private** **double** depth;

// construct clone of an object

Box(Box ob) { // pass object to constructor

width = ob.width;

height = ob.height;

depth = ob.depth;

}

// constructor used when all dimensions specified

Box(**double** w, **double** h, **double** d) {

width = w;

height = h;

depth = d;

}

// constructor used when no dimensions specified

Box() {

width = -1; // use -1 to indicate

height = -1; // an uninitialized

depth = -1; // box

}

// constructor used when cube is created

Box(**double** len) {

width = height = depth = len;

}

// compute and return volume

**double** volume() {

**return** width \* height \* depth;

}

}

**Class 3 – BoxWeight.java:**

// BoxWeight now fully implements all constructors.

**class** BoxWeight **extends** Box {

**double** weight; // weight of box

// construct clone of an object

BoxWeight(BoxWeight ob) { // pass object to constructor

**super**(ob);

weight = ob.weight;

}

// constructor when all parameters are specified

BoxWeight(**double** w, **double** h, **double** d, **double** m) {

**super**(w, h, d); // call superclass constructor

weight = m;

}

// default constructor

BoxWeight() {

**super**();

weight = -1;

}

// constructor used when cube is created

BoxWeight(**double** len, **double** m) {

**super**(len);

weight = m;

}

}

**Result:**

Volume of mybox1 is 3000.0

Weight of mybox1 is 34.3

Volume of mybox2 is 24.0

Weight of mybox2 is 0.076

Volume of mybox3 is -1.0

Weight of mybox3 is -1.0

Volume of myclone is 3000.0

Weight of myclone is 34.3

Volume of mycube is 27.0

Weight of mycube is 2.0