**Main Method()**

*public* *class* generic{

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

*// ---------- Question #1 ---------- //*

*Line*<*Integer*> l1 = new Line(4.0);

        System.out.println(l1.toString());

*// ---------- Question #2 ---------- //*

*Integer* i = 4;

*Double* j = 5.0;

*Rect*<*Integer*, *Double*> rectangle = new *Rect*<*Integer*, *Double*>(i, j);

        System.out.println(rectangle.toString());

*// ---------- Question #3 ---------- //*

*ArrayList*<*Double*> nums = new *ArrayList*<>();

        nums.add(45.0);

        nums.add(55.0);

        nums.add(65.0);

        nums.add(65.0);

        nums.add(75.0);

*Mean*<*Double*> m1 = new Mean();

        System.out.println(m1.calculateMean(nums));

*// ---------- Question #4 ---------- //*

*ArrayList*<*String*> defaultArray = new *ArrayList*<>();

*ArrayList*<*String*> shapes = new *ArrayList*<>();

        shapes.add("Kite");

        shapes.add("Trees");

        shapes.add("Cartoon");

        shapes.add("Sky");

        shapes.add("Hill");

*Draw*<*String*> d1 = new Draw(defaultArray);

*// Method #1*

        System.out.println(d1.isEmpty(shapes));

*// Method #2*

        d1.addItems("Character");

        System.out.println(d1.toString());

*// Method #3*

        d1.drawItem(shapes, 3);

        System.out.println(d1.toString());

    }

}

**Question 1:**

*// Same Data types*

*class* Line <*L*> {

*private* *L* length;

*// Default Constructor*

*public* Line() {

    }

*// Argumented Constructor*

*public* Line(*L* *length*) {

*this*.length = length;

    }

*// Setters*

*public* *void* setLength(*L* *length*){

*this*.length = length;

    }

*// Getters*

*public* *L* getLength() {

        return length;

    }

*public* *String* toString(){

        return length.toString();

    }

}

**Question 2:**

*// Different Data types*

*class* Rect <*L*, *W*> {

*private* *L* length;

*private* *W* width;

*// Default Constructor*

*public* Rect() {

    }

*// Argumented Constructor*

*public* Rect(*L* *length*, *W* *width*) {

*this*.length = length;

*this*.width = width;

    }

*// Setters*

*public* *void* setLength(*L* *length*){

*this*.length = length;

    }

*public* *void* setWidth(*W* *width*){

*this*.width = width;

    }

*// Getters*

*public* *L* getLength() {

        return length;

    }

*public* *W* getWidth() {

        return width;

    }

*public* *String* toString(){

        return length.toString() + " " + width.toString();

    }

}

**Question 3:**

*// Mean Value Calculation*

*class* Mean <*T* *extends* *Number*>{

*// Default Constructor*

*public* Mean() {

    }

*// Mean*

*public* *double* calculateMean(*ArrayList*<*T*> *num*){

*double* finalValue = 0.0;

        for(*int* i = 0; i < num.size(); i++){

            finalValue += num.get(i).doubleValue();

        }

        return finalValue / num.size();

    }

}

**Question 4:**

*class* Draw <*D*>{

*private* *ArrayList*<*D*> newObjects;

*// Argumented Constructor*

*public* Draw(*ArrayList*<*D*> *box*){

*this*.newObjects = box;

    }

*// Setters*

*public* *void* setNewObjects(*ArrayList*<*D*> *box*){

*this*.newObjects = box;

    }

*// Getters*

*public* *ArrayList*<*D*> getNewObjects(){

        return newObjects;

    }

*// Method that checks whether the array is empty or not*

*public* *boolean* isEmpty(*ArrayList*<*D*> *box*){

        if(box.isEmpty()){

            return true;

        }

        else{

            return false;

        }

    }

*// Method to add sepecfic type object to an array*

*public* *void* addItems(*D* *object*){

        newObjects.add(object);

    }

*// Method to draw a specfic drawing from the box*

*public* *void* drawItem(*ArrayList*<*D*> *box*, *int* *index*){

        if(!box.isEmpty()){

            for (*int* i = 0; i < box.size(); i++) {

                newObjects.add(box.get(index));

                System.out.println("Added!");

                break;

            }

        }

    }

*// toString Method*

*public* *String* toString(){

        return newObjects.toString();

    }

}