Object Oriented Programming – SCJ2153

Array of Objects

Associate Prof. Dr. Norazah Yusof

Array of Objects

- We can create an array of objects.

- By combining the advantage of array, and objects, we can build program more clean and better logical organization

- Assume we need to store collection of 50 students, for example, we need to use several different arrays (one for names, one for address, one for matrix number and so forth.) which is very cumbersome and prone to error

- The better way is by creating an **array of students object**. This will result a more concise and easier-to-understand code.

Array of objects declaration

- Declaration of array: 2 ways
- **Way one**: involve 2 steps:

  1) Step 1: Creation of array of type class (to hold objects of that class type).
  - Syntax:

    ```
    ClassName [] ArrayName = new ClassName[ArraySize];
    ```
  - **Example:**

    ```
    Student [] sdntList = new Student[3];
    ```

  2) Step 2: Creation of object in each / particular index of the array.
  - Syntax:

    ```
    ArrayName[index] = new ClassName();
    ```
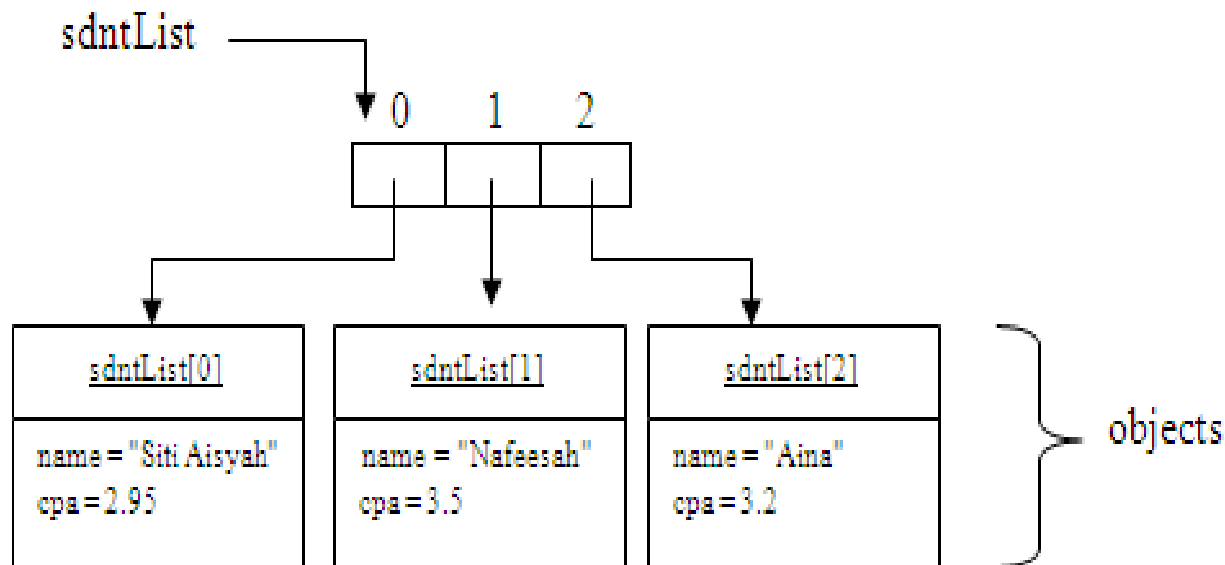 - **Example:**

    ```
    sdntList[0] = new Student("Siti Aisyah", 2.95);

    sdntList[1] = new Student("Nafeesah", 3.5);

    sdntList[2] = new Student("Aina", 3.2);
    ```

Memory Layout

- Memory layout of the one dimensional array of objects:
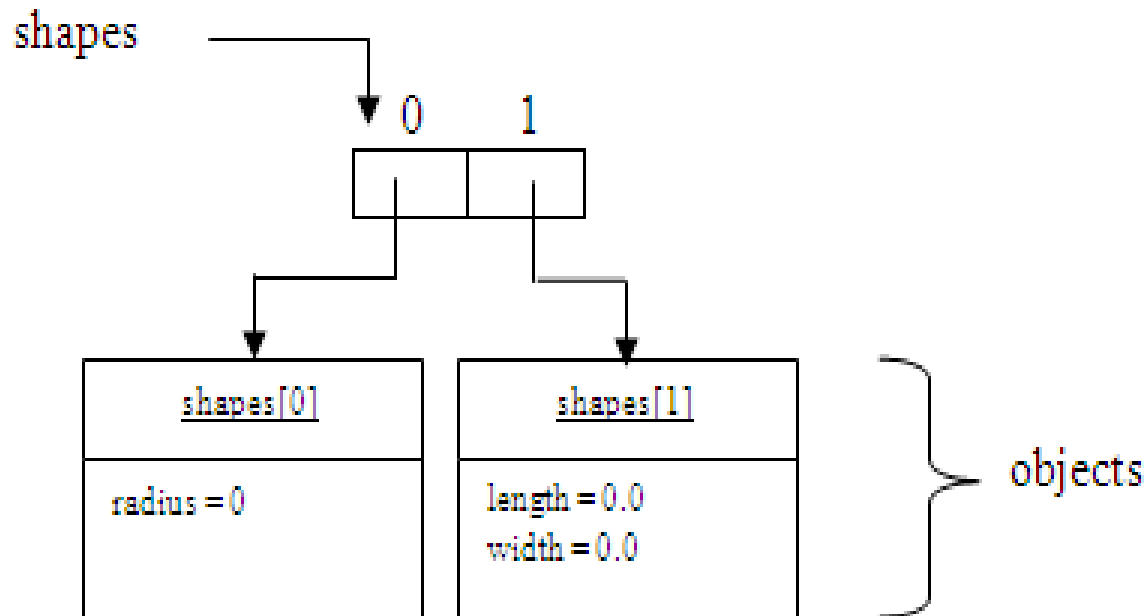
A for loop

to create object for each array element by allowing user input the data from the keyboard.

```
 1 | Scanner input = new Scanner(System.in);
 2 | Student [] sdntList = new Student[3]
 3 | for (int i=0; i<sdntList.length; i++)
 4 | {
 5 |   System.out.println("Please enter a student's name: ");
 6 |   String name = input.nextLine();
 7 |   System.out.println("Please enter a student's cpa: ");
 8 |   double cpa = input.nextDouble();
 9 |   sdntList[i] = new Student(name, cpa);
10 | }
```

Declare and Initialize

- An array of objects can also be declared, created and initialized.

```
Shape [] shapes = {new Circle(), new Rectangle()};
```
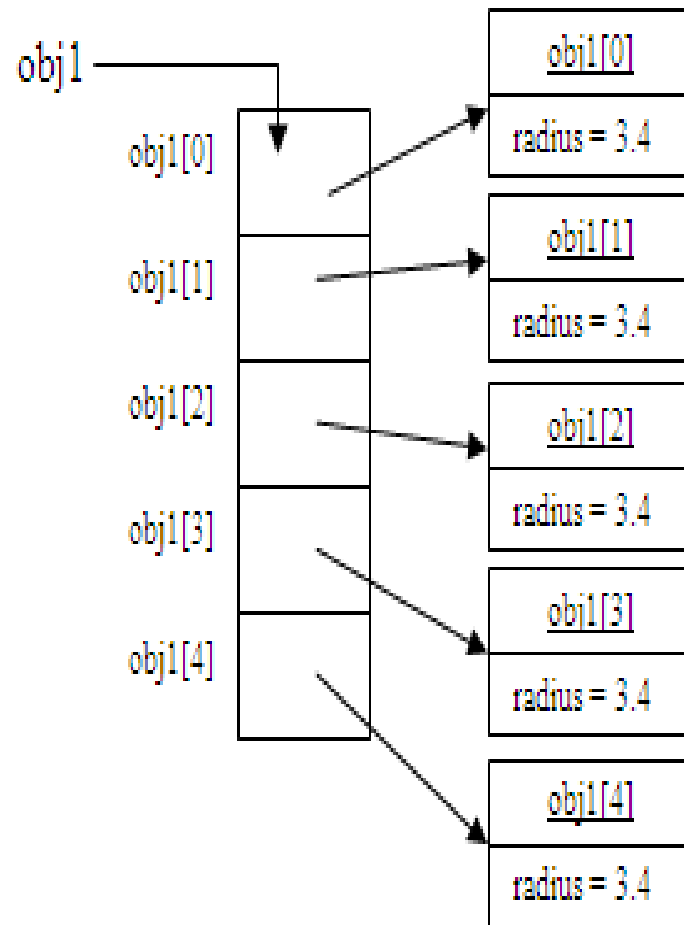
Enhanced-for loop

- Java also supports the enhanced-for loop for accessing array of objects.

```
1  Float [] floatObj = {4.1f, 5.2f, 6.3f};
2  float [] floatPrim = {4.1f, 5.2f, 6.3f};
3  for(float y:floatPrim) {
4    System.out.print(y + " " );
5  }
6  System.out.println();
7  for(Float fp: floatObj){
8    System.out.print(fp + " " );
9  }
```

Memory layout

for loop vs enhanced-for loop

- conventional for loop and enhanced for loop

Conventional for loop	Enhanced for loop
```Circle [] obj1 = new Circle[3];``` ``` ``` ```for (int i=0; i<3; i++) {``` ``` obj1[i] = new Circle();``` ``` obj1[i].setRadius(3.4);``` ```}``` ``` ``` ```for (int i=0; i<obj1.length; i++)``` ``` obj1[i].displayInfo();```	```Circle [] obj1 = new Circle[3];``` ``` ``` ```for (Circle i:obj1){``` ``` i = new Circle();``` ``` i.setRadius(3.4);``` ``` i.displayInfo();``` ```}```