Object Oriented Programming (OOP)

Mohamed Ezz

Lecture 5

Review

- ➢Draw the class diagram showing has-a relationship for:
  - ✓ Family, Father, Mother, Son, Daughter classes
  - ✓ Building, Address, Contact classes
  - ✓LinkedList, Node
- ➢Which of the access specifiers are suitable for the following :
  - ✓ password in customer class
  - ✓GetName in customer class
  - ✓PI in Math class
- ➢Write signature of two or more overloaded methods for the following classes:
  - ✓ constructors of Student class
  - ✓ constructors of Point class
  - ✓drawDiagram for Paint class (for point, rectangle, circle, triangle)
  - ✓calculatePaint for Paint class (for point, rectangle, circle, triangle)
- ➢Write static method & variable for the counter of Student class

Lecture Objectives

- ✓ Review Array in Java
- ✓ Understand how to create Array of user-defined Objects
- ✓ Understand how to analysis & design advanced Composition
- ✓ Understand ArrayList & HashMap built-in collection/container java classes
- ✓ Practice composition example using built-in collection java classes

Review : Arrays

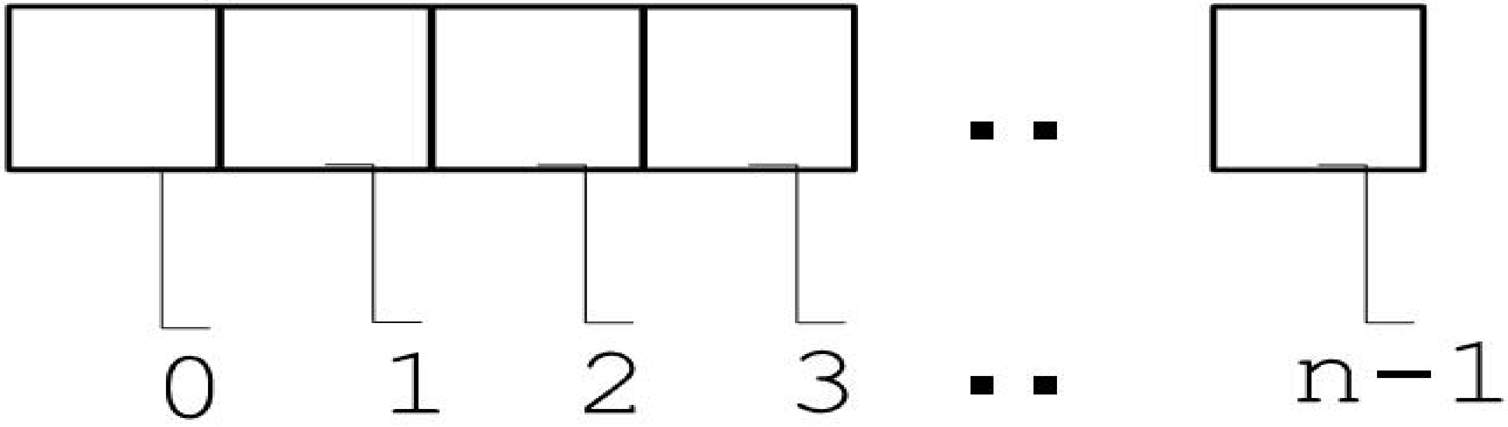
An array is an indexed list of values.

You can make an array of any type

- int, double, String, etc..

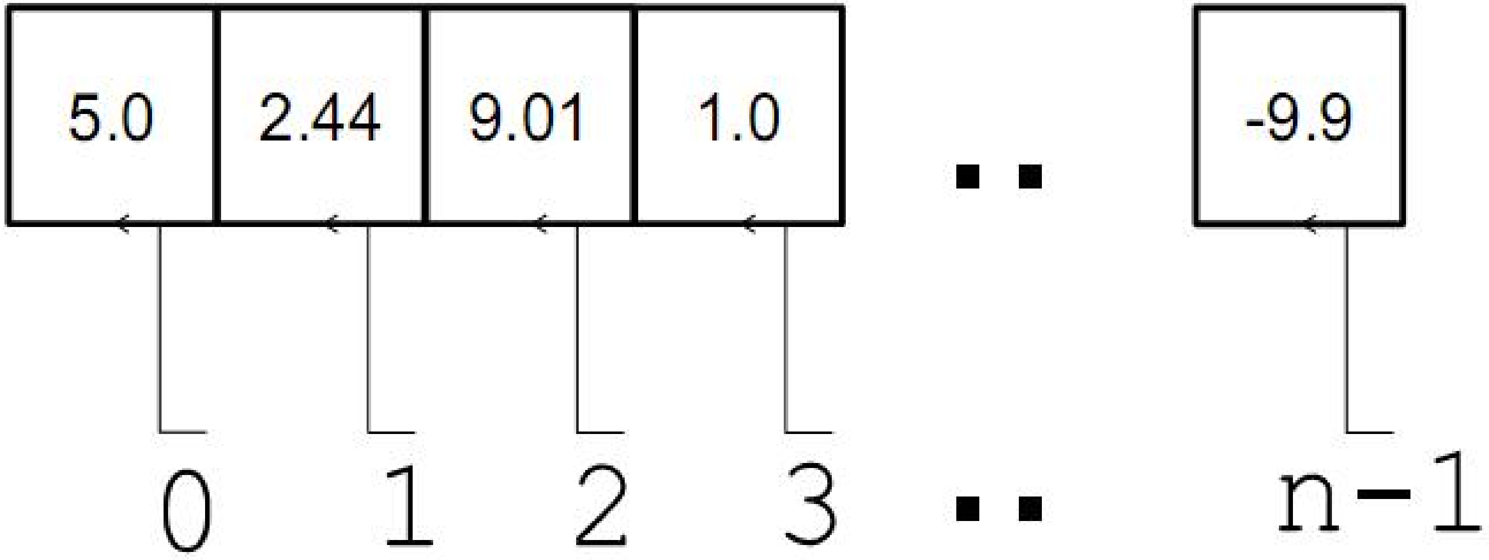
All elements of an array must have the same type.

Arrays



Arrays

Example: double [ ]



5.0 2.44 9.01 1.0 .. -9.9

0 1 2 3 .. n-1

Arrays

The index starts at zero and ends at length-1.

Example:

```
int[] values = new int[5];
values[0] = 12; // CORRECT
values[4] = 12; // CORRECT
values[5] = 12; // WRONG!! compiles but
                // throws an Exception
                // at run-time
```

Have a demo with runtime exception

Arrays

An array is defined using TYPE[].

Arrays are just another type.

```
int[]   values;   // array of int

int[][] values;   // int[] is a type
```

Arrays

To create an array of a given size, use the operator `new` :

```
int[] values = new int[5];
```

or you may use a variable to specify the size:

```
int size = 12;
int[] values = new int[size];
```

Array Initialization

Curly braces can be used to initialize an array.

It can ONLY be used when you declare the variable.

```
int[] values  = { 12, 24, -23, 47 };
```

Accessing Arrays

To access the elements of an array, use the `[]` operator:

```
values[index]
```

Example:

```
int[] values  = { 12, 24, -23, 47 };
values[3] = 18;           // {12,24,-23,18}
int x = values[1] + 3;  // {12,24,-23,18}
```

The *length* variable

Each array has a `length` variable built-in that contains the length of the array.

```
int[] values =  new int[12];
int size =      values.length; // 12

int[] values2 = {1,2,3,4,5}
int size2 =     values2.length; // 5
```

String arrays

A side note

```
public static void main (String[] arguments){
    System.out.println(arguments.length);
    System.out.println(arguments[0]);
    System.out.println(arguments[1]);
}
```

Looping through an array

Example 1:

```
int[] values = new int[5];

for (int i=0;  i<values.length; i++) {
   values[i] = i;
   int y = values[i] * values[i];
   System.out.println(y);
}
```

Looping through an array

Example 2:

```
int[] values = new int[5];
int i = 0;
while (i < values.length) {
   values[i] = i;
   int y = values[i] * values[i];
   System.out.println(y);
   i++;
}
```

Other way to access array

- Traditional way to access the array

```
      String[] names  = {"Ahmed","Mohamed", "Aly"};
      for ( int i=0;i< names.length;i++){
            String s= names[i];
             System.out.println( s);
            }
```

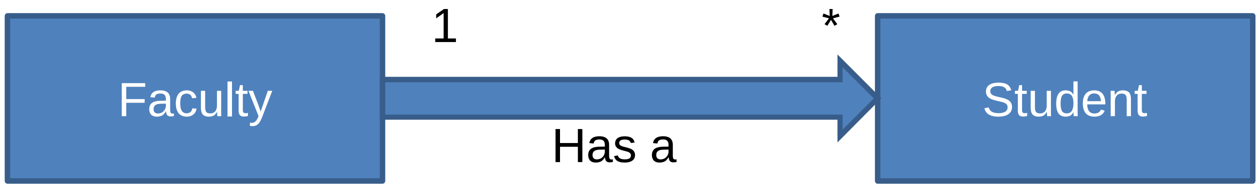
- Instead of looping & access array using index, use the following form:

```
 String[] names  = {"Ahmed","Mohamed", "Aly"};
       for(String s: names){
           System.out.println( s);
        }
```

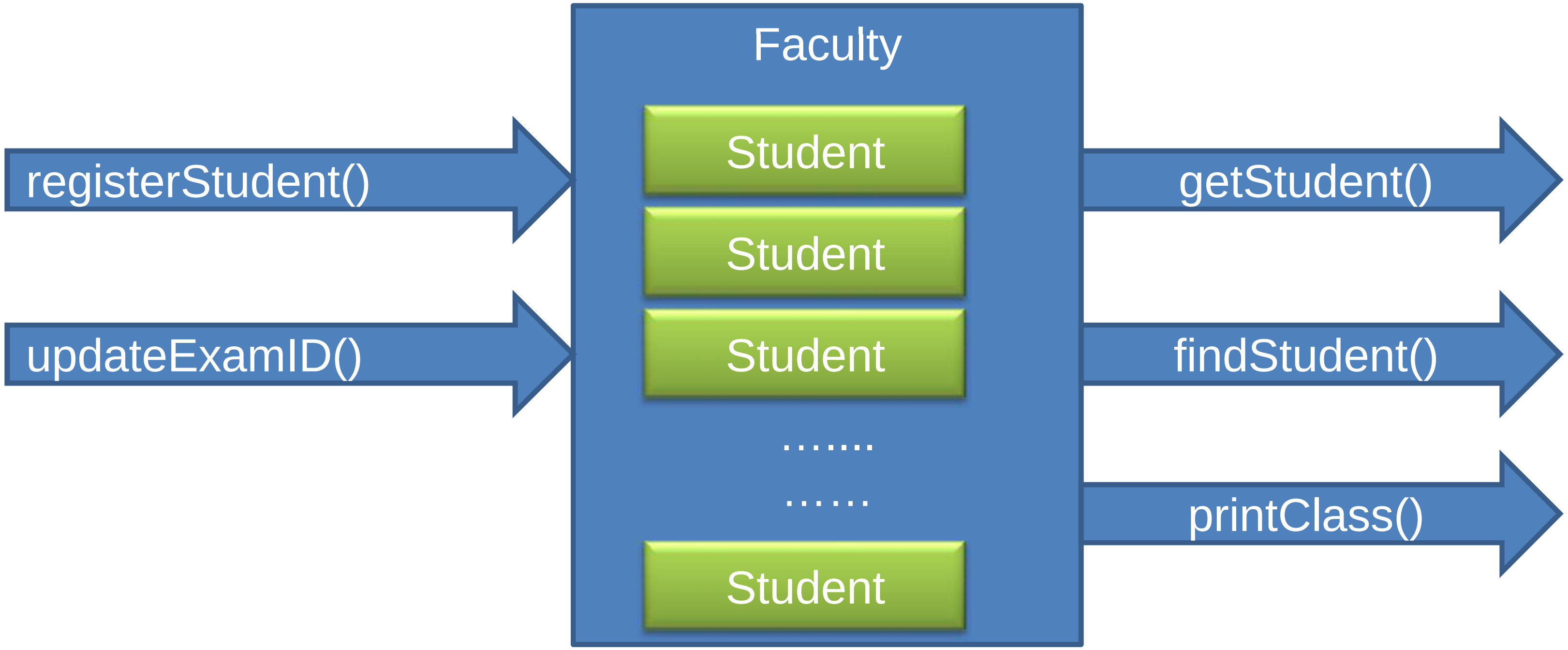
Array of Object “Object”

- Facebook
- Gmail/Hotmail/Yahoo
- Forum: has array of Members
- Faculty: has array of Students

Faculty & Student Relations



Faculty Class has member variable student array (of type student[])



Faculty & Student Classes

Student.java

```
class Student{
        String firstName;
         String LastName;
         int classNo;// 1,2,3,4
         int examID;
        public Student(String fName, String lName, int classNo){
            firstName = fName
            lastName = lName;
            this.classNo=classNo;
        }
        public void print(){
            System.out.println(firstName + “ ” + lastName+ “ ID=”+  examID);

        }
}
```

Faculty & Student Classes

Faculty.java

```
class Faculty {
   private Student[]  students;
   private int index;
   private int max;
   public Faculty(int NoOfStudent){
     students = new Student[NoOfStudent];
     index=0;
     max= NoOfStudent;
   }
   public void registerStudent(Student s){
     if (index <max){
       students[index]= s;
       index++;
    }
    else
     System.out.println("Faculty is full");
   }
```

```
   public Student getStudent(String name){
     for (int i=0;i<index;i++){
       String sName = students[i].firstName;
        if (sName.equals(name))
            retrun students[i];
     }
     return Null;
   }

public void printStudents(int cID){
    for (int i=0;i<index;i++){
        if (students[i].classNo= cID)
              students[i].print();
    }
}
```

Test Faculty & Student

TestFaculty.java

```
class TestFaculty {
        public static void main(String arg[]){
                Faculty engFaculty = new Faculty (100);
                Student    student1 = new Student(“Amal”,”Amer”,1);
                Student    student2 = new Student(“Wael”,”Aziz”,2);
                Student    student3 = new Student(“”Sayed,”Yousef”,3);
                 engFaculty. registerStudent(student1);
                 engFaculty. registerStudent(student2);
                 engFaculty. registerStudent(student3);
                 engFaculty. registerStudent(new Student(“Moh”,”Ezz”,4) );
                 engFaculty.updateExamID();
                 Student student = engFaculty getStudent(“Wael”);
                student.classNo=1;
                 engFaculty.printStudent(1);
        }
}
```

ArrayList & HashMap

ArrayList & HashMap

Limitations of traditional Array :

✓ complexity of delete operation : deleteStudent

✓Search operation is low speed: FindStudent (image search for user in facebook)

✓Fixed list of items which is not scalable

This is the time to learn ArrayList & HashMap classes of java.util to overcome these limitations

ArrayList

- Class in **java.util** java package
- Support list of any object type (such as String, Point, Rectangle…)
- Declaration:
  - ArrayList var_listName = new ArrayList ();
  - e.g.
  - ArrayList   names = new ArrayList ();
  - ArrayList points= new ArrayList ();
- Add Element :
  - **void add(Object type):** Appends the element to the end of this vector.
  - e.g.
    - names.**add(“**Omar**”);**
    - points.**add(**p1**);**
- Get Element :
  - **Object type get(int index):** Returns the object at the specified index
  - e.g.
    - String name= names.**get(2);**
    - Point p= points.**get(**1**);**

ArrayList Example

TestArrayList.java

```
Class TestArrayList{
  public static void main(String arg[])
      ArrayList   friends = new ArrayList ();
      friends.add("Ahmed");
      friends.add("Mohamed");
      friends.add("Ali");

      for(int i=0;i<friends.size();i++){
         System.out.println(friends.get(i));
      }
      friends.remove (1);

      for(int i=0;i<friends.size();i++){
         System.out.println(friends.get(i));
      }
  }
}
```

Other way to access array

```
•ArrayList friends = new ArrayList();
       friends.add("Ahmed");
       friends.add("Mohamed");
       for ( int i;i< friends .size();i++){
             String f= friends [i];
              System.out.println( f);
      }
```

Instead of looping & access array or ArrayList using index, use the following form:

```
      ArrayList friends = new ArrayList();
       friends.add("Ahmed");
       friends.add("Mohamed");
       for(String f: friends){
           System.out.println(f);
       }
```

Other way to access array

```
ArrayList points = new ArrayList();
points.add(new Point(1,1));
points.add(new Point(2,3));
for(Point p: points){
    System.out.println(“x=” +p.x + “ y =” +p.y);
}
```

Syntax:

```
for(Object_type variable_name : arrayName){
        // inside this for you will use variable_name directly

    }
```

Notes: array_name is either array name or ArrayList object ref.

Faculty Class using ArrayList

```
Import java.util.*;
class Faculty {
   private ArrayList  students;
   public Faculty(){
     students = new ArrayList();
  }
   public void registerStudent(Student s){
       students.add(s);
   }
  public Student getStudent(String name){
     for ( Student s: students){
         if (s.firstName.equals(name))
            return s;
     }
      return  null
   }
```

```
public void printStudent(int cID){
 for ( Student s: student){
     if (s.classNo == cID)
             s.print();
    }
}

public void updateExamID(){
  int  c1=0,c2=0,c3=0,c4=0;
  for ( Student s: student){
    switch (s.classNo){
             case 1: c1++; s.examID=c1;break;
             case 2: c2++; s.examID=c2; break;
             case 3: c3++; s.examID=c3; break;
             case 4: c4++; s.examID=c4; break;
    }
  }
}
```

Test Faculty with ArrayList

```
class TestFaculty {
                public static void main(String arg[]){
                                Faculty engfaculty = new Faculty ();
                                Student    student1 = new Student(“Amal”,”Amer”,1);
                                Student    student2 = new Student(“Wael”,”Aziz”,2);
                                Student    student3 = new Student(“”Sayed,”Yousef”,3);
                                 engfaculty. registerStudent(student1);
                                 engfaculty. registerStudent(student2);
                                 engfaculty. registerStudent(student3);
                                 engfaculty. registerStudent(new Student(“Moh”,”Ezz”,4) );
                                 engfaculty updateExamID();
                                 Student student = engfaculty . getStudent(“Wael”);
                                student.classNo=1;
                                 engfaculty.printStudent(1);
                }
}

// Any change on consumer side?
```

HashMap

- Class in **java.util** java package
- Support list of any object type (such as String, Point, Rectangle…) with **key**
- Declaration:

  HashMap   var_listName = new HashMap();
  e.g.
  HashMap mails = new HashMap();
  HashMap points= new HashMap();
- Add Element :

  void put(**key, object** **): put** the element to the table.
  e.g.
  mails.**put(**“Omar”**, “omar@gmail.com”);**
- Get Element :

  Object **get (key):** Returns the object corresponding to key
  e.g.
  String emal= mails.**get(“Omar”);**

This is faster than getStudent(String name) **in Arraylist version**

HashMap Example

```
import  java.util.*;
Class TestHashMap{
  public static void main(String arg[])
  {
     HashMap contacts = new HashMap();
      contacts.put("Ahmed", "ahmed@gmail.com");
      contacts.put("Mohamed","mohamed@gmail.com");
      contacts.put("Ali","ali@gmail.com");

      System.out.println(  contacts.get("Ali")  ) ;
 }
}
```

Faculty Class using HashMap

```
Import java.util.*;
class Faculty {
    private HashMap students;
    public Faculty(){
       students=new HashMap();
   }
    public void registerStudent(Student s){
          students.put(s.name, s);
    }
   public Student getStudent(String name){
     return  students.get(name);
    }
```

```
....
```

Test Faculty with HashMap

```
class TestFaculty {
        public static void main(String arg[]){
                Faculty faculty = new Faculty ();
                Student    student1 = new Student(“Amal”,”Amer”,1);
                Student    student2 = new Student(“Wael”,”Aziz”,2);
                Student    student3 = new Student(“”Sayed,”Yousef”,3);
                faculty. registerStudent(student1);
                faculty. registerStudent(student2);
                faculty. registerStudent(student3);
                faculty. registerStudent(new Student(“Moh”,”Ezz”,4) );
                faculty.updateExamID();
                 Student student = faculty. getStudent(“Wael”);
                student.classNo=1;
                faculty.printStudent(1);
        }
}

// Any change on consumer side?
```

Homework

- UML of **Facebook** Application
  - List of User Profiles
    - User profile
      - Username
      - Name
      - E-mail
      - Mobile
      - **Wall** : list posts from Me & my friends
        - **Post**:
          - Description
          - Media (image/video)
          - List of comments
          - List of like/dislike
      - **photo** : List of images
      - **Friends**: list of users

Homework

Design an email system with define Classes relationship