SENG1110/SENG6110
Object Oriented Programming

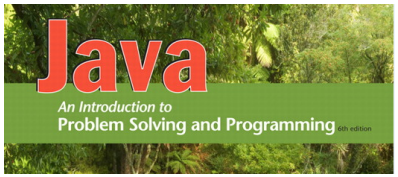
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

FACULTY OF ENGINEERING AND BUILT ENVIRONMENT

www.newcastle.edu.au

Lecture 9

Arrays – part II



Outline

- Previously…
  - Array Basics
  - Arrays in Classes and Methods
- Now…
  - Sales example - physical size x logical size
  - Arrays of class type
    - Person example
    - Agency example
    - Tutor example
    - Grade report example

Resize an array

- Suppose the program instantiate an array with size 10
- After some time, the array can be full…
- How to resize the array?

Resize an array

- **The copy2 method…**

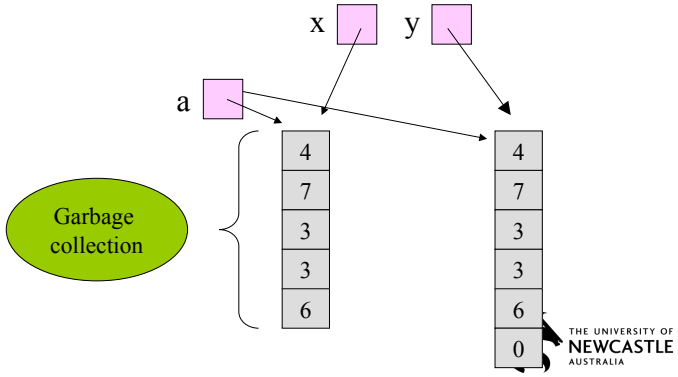
```
public int[] resize(int[] x)
   {
        int[] y = new int[x.length+n];
        for(int i=0; i< x.length; i++)
               y[i] = x[i];
        return(y);
   }
```

New size. It can be

x.length+1
x.length*2
…

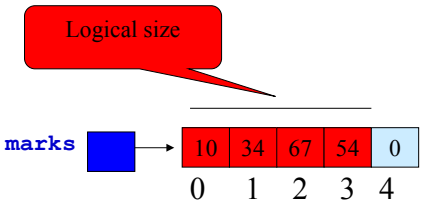
Resize an array


```
public int[] resize(int[] x, int plus)
{
      int[] y = new int[x.length+plus];
      for(int i=0; i< x.length; i++)
            y[i] = x[i];
      return(y);
}
...
a = resize(a,1)
```



Physical size x logical size

- Physical size = number of cells in array
- Logical size = number of values currently stored and used by the program



- But…how to manage the logical size?

Sales example

- Let's see an example…
  - Using array of integers `sales`
  - Using just one class – Sales.java
  - How to manage the logical size

Sales example


```
import java.util.*;
public class Sales
{
  public static void main (String[] args)
  {
        final int MAX = 5;
        int[] sales = new int[MAX];
        int total=0;
        int choice;
```

`total` = array logical size
`sales.length` = physical size

Sales example


```
showSelection();
choice = console.nextInt();
while(choice != 9)
{
   switch(choice)
   {
        case 1:  addSale(sales);
                 break;
        case 2:  position = console.nextInt();
                 deleteSale(position,sales);
                 break;
        case 3:  System.out.println("average = "+averageSale(sales));
                 break;
        case 9:  break;
                 default: System.out.println("Invalid Selection");
    }//end switch
   showSelection();
   choice = console.nextInt();
}
```


Sales example


```
public static void showSelection()
{
   System.out.println("Select and enter");
   System.out.println("1 - add a new sale value");
   System.out.println("2 - delete a sale ");
   System.out.println("3 - average");
   System.out.println("9 - exit");
}
public static void addSale(int[] s)
{
   Scanner console = new Scanner(System.in);
   s[total]= console.nextInt();
   total++;
}
```


Sales example


```
public static void deleteSale(int position, int[] s)
{
   for(int i=position; i<total-1; i++)
        s[i] = s[i+1];
   total--;
}
public static double averageSale(int[] s)
{
   double average=0;
   for(int i=0; i<total; i++) {
        average += s[i];
   }
   return(average/total);
}
```


Sales example

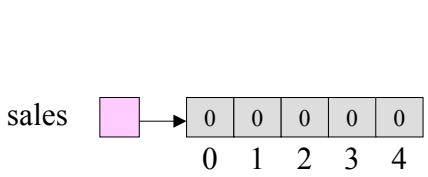
- Let's see each method

```
public static void addSale (int[] s)
{
   s[total]= console.nextInt();
   total++;
}
```

- `total` starts with 0

Sales example


```
s[total] = console.nextInt();
total++;
```

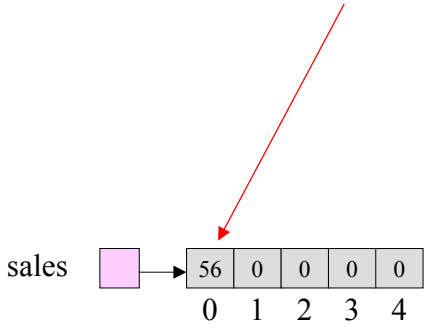


total

0

Sales example


```
s[total] = console.nextInt();
total++;
```

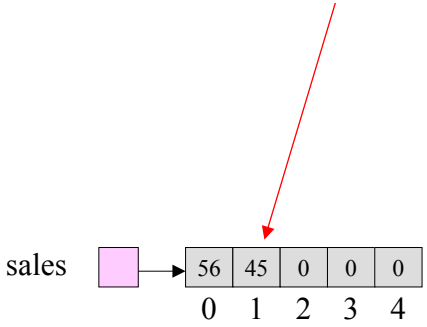


total

1

Sales example


```
s[total] = console.nextInt();
total++;
```

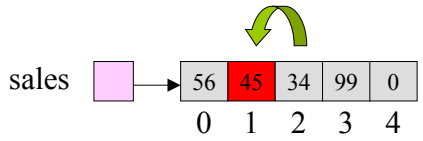


total

2

Sales example


```
public static void deleteSale(int position, int[] s)
{
   for(int i=position; i<total-1; i++)
        s[i] = s[i+1];
   total--;
}
```

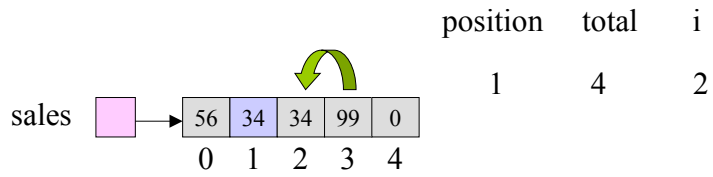


position total i

1 4 1

Sales example

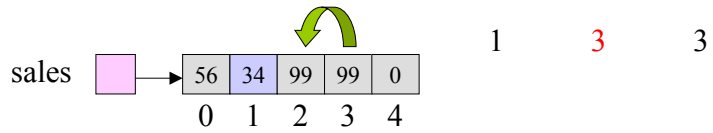

```
public static void deleteSale(int position, int[] s)
{
   for(int i=position; i<total-1; i++)
        s[i] = s[i+1];
   total--;
}
```



Sales example


```
public static void deleteSale(int position, int[] s)
{
   for(int i=position; i<total-1; i++)
        s[i] = s[i+1];
   total--;
}
```

total is the array logical size

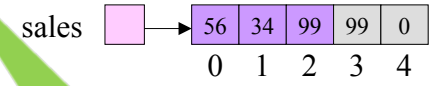


Sales example


```
public static double averageSale(int[] s)
{
   double average=0;
   for(int i=0; i<total; i++) {
        average += s[i];
   }
   return(average/total);
}
```

total 3

average 63



See that it will go until position 2, ie, it will sum 3 numbers (56+34+99), since the logical size is 3

Sales example

- The initial size of the array is MAX=5.
- If the number of sales added is more than 5, what will happen?
- How to fix?

Adding a method resizeArray
Try to do it.

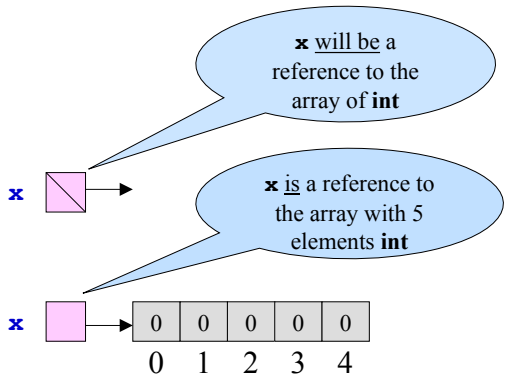
Arrays of primitive types

- **Primitive** type:
  - declaration

```
int[] x;
```

  - Instantiation

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



  - Notice that
    - `x` is an array; `x` is an reference to
    - `x[3]` if the position 3 of the array. You can do, for example, `x[3]=5;`

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Arrays of class types

- For arrays of non-primitive types you must create the array first, then load it with objects

```
final int MAX = 5;
Person[] people = new Person[MAX];

for (int i=0; i<MAX; i++)
      people[i] = new Person();
```

- Array cells are null when the array is instantiated
- New objects must then be assigned to the cells

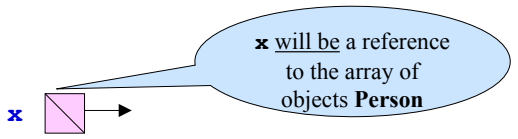
- Let's see what happen…

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Arrays declaration and instantiation

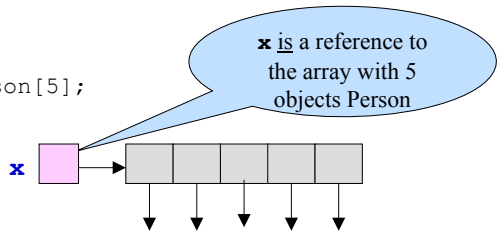
- **Reference** type:
  - declaration

```
Person[] x;
```



  - Instantiation

```
x = new Person[5];
```



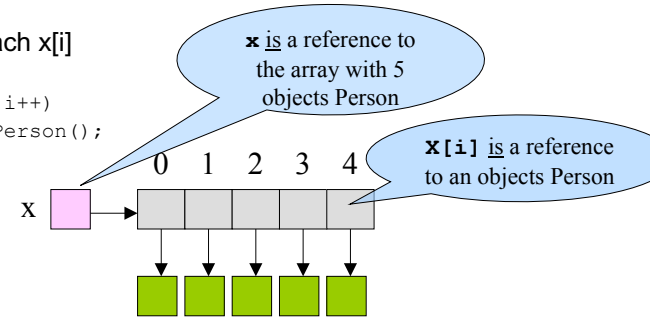
  - Note that
    - `x` is an array; `x` is an reference to…
    - `x[3]` if the position 3 of the array. In this case, `x[3]` is an reference to…if fact, it will be a reference to…we need to instantiate…let's to continue…

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Arrays of non-primitive types

- Instantiation of each x[i]

```
for (i=0; i<5; i++)
   x[i] = new Person();
```



- `x` is an array; `x` is an reference to…
- `x[3]` if the position 3 of the array. In this case, `x[3]` is an reference to…
- How to access to the instance variables and/or methods from Person object? Let's to continue…

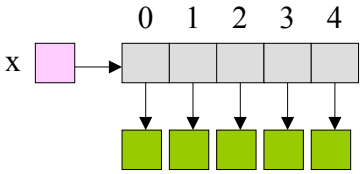
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Arrays of class types

– Using the objects' methods/variables

```
x[0].name = "Andre"
x[1].age = 34
x[3].setName(s) or x[3].setName("Andre")
x[4].setAge(34)
a = x[3].getAge();
```

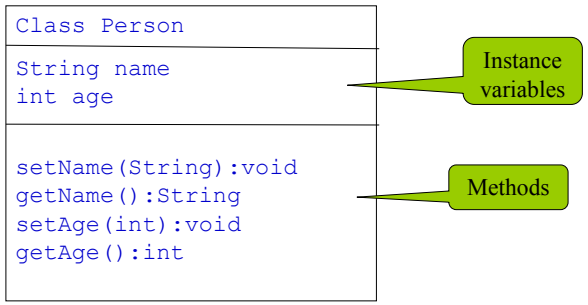
If the instance variables (name and age) were declare private, these lines will not work



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 1 – Person class

- Suppose class Person:



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 1 – Class Person


```
public class Person
{
    private String name;
    private int age;

    public Person()
    {
        name = "";
        age = 0;
    }
    public void setName(String name)
    {
        this.name = name;
    }
    public String getName()
    {
        return name;
    }
    public void setAge(int newAge)
    {
        age = newAge;
    }
    public int getAge()
    {
        return age;
    }
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 1 – Using class Person / no array

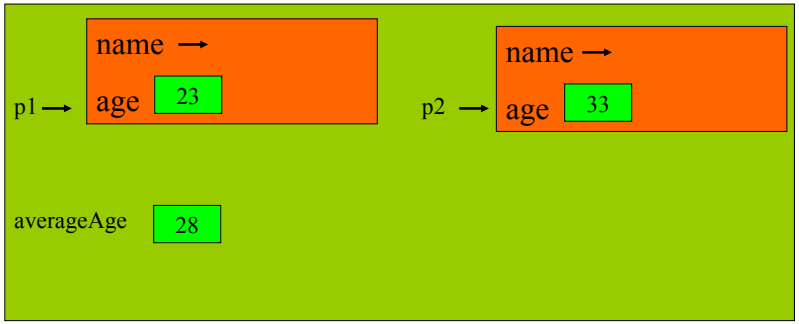
UsePerson.java

```
public class UsePerson
{
   public static void main(String[] args)
   {
         Scanner console = new Scanner(System.in);
         private Person p1,p2;
         private int averageAge;

         p1 = new Person();
         p2 = new Person();

         p1.setAge(console.nextInt());
         p2.setAge(console.nextInt());

         averageAge = (p1.getAge() + p2.getAge())/2;
         System.out.println(averageAge);
   }
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 1 – Using class Person / no array

- in UsePerson we have two objects



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 2 – Using class Person / with array

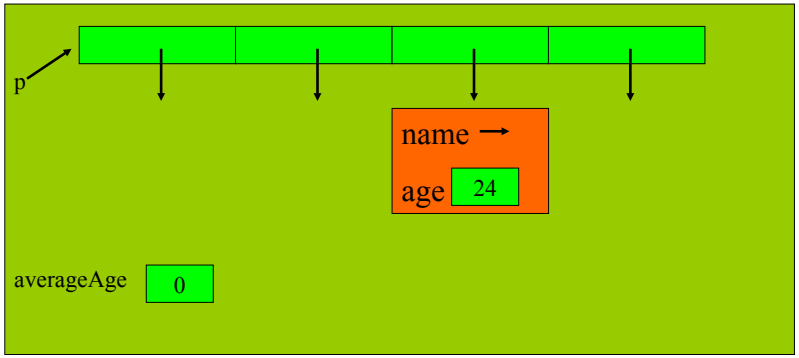
UsePersonA.java

```
public class UsePersonA
{
   public static void main(String[] args)
   {
         Scanner console = new Scanner(System.in);

         Person[] p = new Person[4];
         int averageAge;

         p[2] = new Person();

         p[2].setAge(console.nextInt());

   }
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 2 – Using class Person / with array

- UsePersonA has an array of Person



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

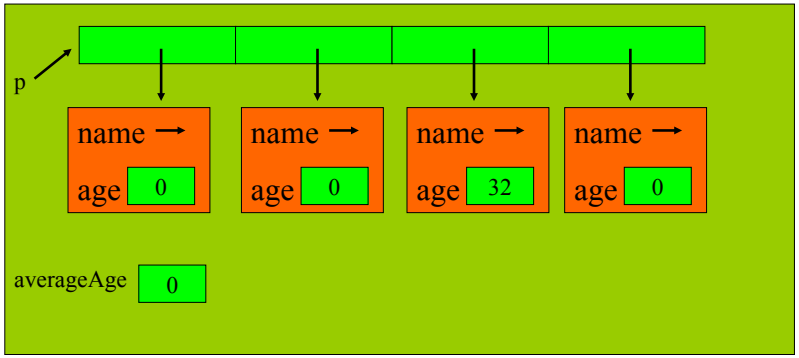
Example 2 – Using class Person / with array

UsePersonA.java

```
public class UsePersonA
{
   public static void main(String[] args)
   {
         Scanner console = new Scanner(System.in);

         Person[] p = new Person[4];
         int averageAge;

         for (int i=0; i<4; i++)
                  p[i] = new Person();

         p[2].setAge(console.nextInt());

   }
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 2 – Using class Person / with array



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 3 – Using class Person / with array

- Let's calculate the average age of the elements in the array
  - Version 1 – without method
  - Version 2 – with method

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 3 – Using class Person / with array

Average Version 1

```
public class UsePersonB
{
    public static void main(String[] args)
    {
          private Person[] p = new Person[4];
          private int averageAge;

          for (int i=0; i<4; i++)
          {
                  p[i] = new Person();
                  p[i].setAge(console.nextInt());
          }
          for(int i=0; i<4; i++)
                  averageAge+=p[i].getAge();
          System.out.println(averageAge/a.length);
    }
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example 3 – Using class Person / with array

Average Version 2

```
public class UsePersonB
{
    public static void main(String[] args)
    {
          Person[] p = new Person[4];
          for (int i=0; i<4; i++)
          {
                  p[i] = new Person();
                  p[i].setAge(console.nextInt());
          }
          System.out.println(calculateAverageAge(p));
    }
    public static int calculateAverageAge(Person[] x)
    {
          int averageAge;
          for(int i=0; i<4; i++)
                  averageAge+=x[i].getAge();
          return(averageAge);
    }
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example - Agency with array

- First, let's remember the example
  - There are 3 classes.
    - Person class - represents one person
    - Couple class - represents 2 people
    - AgencyInterface class – interface with the user
    - The main method will be in Agency interface.

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Person class


```
public class Person
{
        private String name;
        private int age;

    public Person()
    {
        name = "";
        age = 0;
    }
    public void setName(String newName)
    {
        name = newName;
    }
    public String getName()
    {
        return name;
    }
    public void setAge(int newAge)
    {
        age = newAge;
    }
    public int getAge()
    {
        return age;
    }
}
```

Class name: Person

Instance variables

Constructor method: Person

More methods: setName getName setAge setAge

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Couple class


```
public class Couple
{
    private Person he,she;

    public Couple()  {
        he  = new Person();
        she = new Person();
    }
    public void setData(int option, String name, int age)
        if (option==1) setData1(she,name,age);
        else           setData1(he,name,age);
    }
    private void setData1(Person p, String name, int age)    {
            p.setName(name);
            p.setAge(age);
    }
    public String test()
    {
        if (she.getAge() < he.getAge()) return("GOOD FOR "+he.getName()+"!");
        else                            return("GOOD FOR "+she.getName()+"!");
    }
}
```

Class name: Couple

Instance variables

Constructor method: Couple

More methods: setData, setData1 test

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

AgencyInterface class


```
import java.util.*;

public class AgencyInterface
{
    public static void main (String[] args)
    {
        Scanner console = new Scanner(System.in);
        Couple c = new Couple();
        int    herAge,hisAge,end;
        String herName,hisName;
```

Class name: AgencyInterface

variables: Object **c** from Couple class

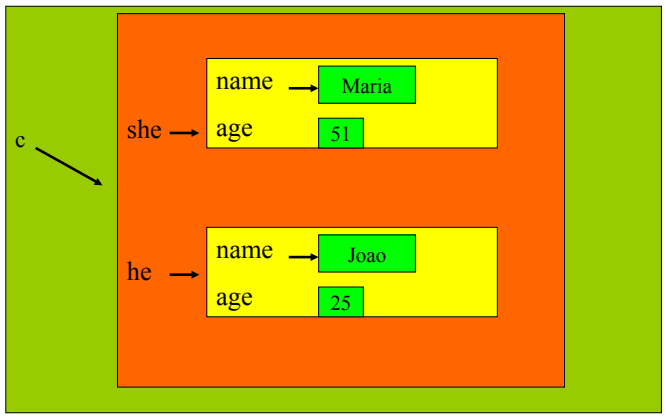
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

AgencyInterface class


```
do {
            System.out.print("her name: ");   herName = console.next();
            System.out.print("her age: ");    herAge = console.nextInt();
            System.out.print("his name: ");   hisName = console.next();
            System.out.print("his age: ");    hisAge= console.nextInt();

            c.addData(1,herName,herAge);
            c.addData(2,hisName,hisAge);

            System.out.println("********************");
            System.out.println(c.test());
            System.out.println("********************");

            System.out.print("Quit? (0)yes (1)no: ");
            end = console.nextInt();
            }
        while (end!=0);
        }
}
```


Example - Agency without array

- Suppose the user enter some data. So, you have:



Example Agency – with array

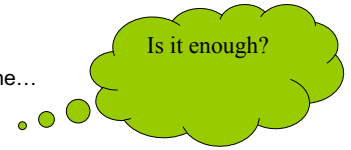
- After main()…go to constructor in AgencyInterface

```
public AgencyInterface(){
        c = new Couple();
        setTitle("Agency");
       }
```

- Instantiate the Couple object c… call the constructor inside Couple class

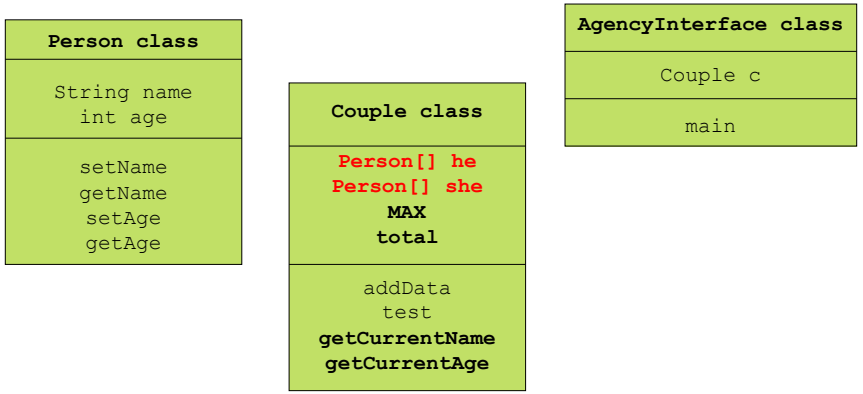
```
public Couple(){
       he  = new Person[MAX];
       she = new Person[MAX];
       total = 0;
       }
```

- Instantiate the Array of Person object he and she…

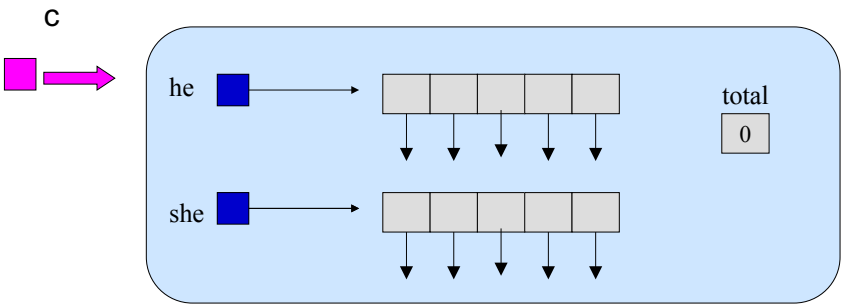


Example Agency – with array

- What we have…



Just the object c...



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

AgencyInterface class


```
import java.util.*;

public class AgencyInterface
{
 Scanner console = new Scanner(System.in);

  Couple c = new Couple();

  public static void main (String[] args)
  {
        int      choice,position;
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

AgencyInterface class


```
showSelection();
choice = console.nextInt();
while(choice != 9)  {
    switch(choice) {
          case 1:   addCouple();
                    break;
          case 2:   position = console.nextInt();
                    testCouple(position);
                    break;
          case 3:   position = console.nextInt();
                    displayCouple(position);
                    break;
          case 9:   break;
                    default: System.out.println("Invalid Selection");
       }//end switch
    showSelection();
   choice = console.nextInt();
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

AgencyInterface class


```
public static void showSelection()
{
    System.out.println("Select and enter");
    System.out.println("1 - add a new couple");
    System.out.println("2 – test a couple");
    System.out.println("3 – display couple");
    System.out.println("9 - exit");
}
public static void addCouple()
{
    String   herName,hisName;
    int      herAge,hisAge;

    System.out.print("her name: ");  herName = console.next();
    System.out.print("her age: ");   herAge  = console.nextInt();
    System.out.print("his name: ");  hisName = console.next();
    System.out.print("his age: ");   hisAge  = console.nextInt();

    c.addData(herName,herAge,hisName,hisAge);
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

AgencyInterface class


```
public static void testCouple (int position)
{
   System.out.println(c.test(position));
}
public static double displayCouple(int position)
{
   System.out.println(c.display(position));
}
}
```


Couple class


```
public class Couple
{
    final private int MAX = 5;
    private Person[] he,she;
    private int total;

public Couple()
{
        he   = new Person[MAX];
        she = new Person[MAX];
        total = 0;
}
```

Instance variables

Constructor method: Couple

Couple class


```
public void addData(String name1, int age1, String name2, int
   age2)
{
    she[total] = new Person();
    he[total] = new Person();
    she[total].setName(name1);
    she[total].setAge(age1);
    he[total].setName(name2);
    he[total].setAge(age2);
    total++;
}
```

It is necessary instantiate each array position

We call setName and setAge twice. We can do this inside a method…Let's see

Couple class

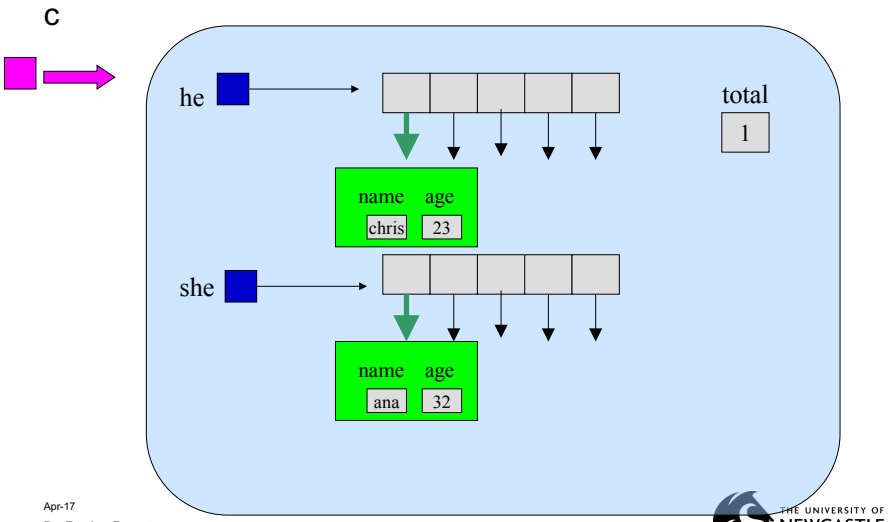

```
public void addData(String name1, int age1, String name2, int age2)
{
   she[total] = new Person();
   he[total] = new Person();
   setData1(she[total],name1,age1);
   setData1(he[total],name2,age2);
   total++;
}
private void setData1(Person p, String name, int age)
{
   p.setName(name);
   p.setAge(age);
}
```

Goes to Person

Couple class

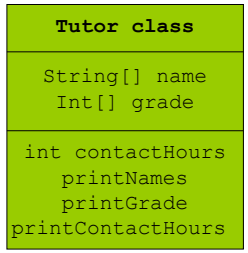

```
public String test(int pos)
{
       if (position!=-1)  {
        if (she[pos].getAge()<he[pos].getAge()) return("GOOD FOR+he[pos].getName()+"!");
        else                                    return("GOOD FOR"+she[pos].getName()+"!");
       }
       return "error";
}
public String display(int position)
    {
        if (position!=-1)
          return("she:"+she[position].getName()+" "+she[position].getAge()+"\n"+
                 "he:"+he[position].getName()+" "+he[position].getAge());
        else
          return("error");
    }
}
```

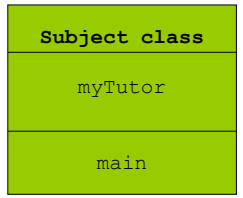

The object c after we called addData once...



Example Tutor

- This example works with arrays of primitive types
- We use these arrays to instantiate an object





Tutor class


```
public class Tutor
{
     String[] name = new String[4];
     int[] grade = new int[4];
     int contactHours;

     public Tutor(String[] nm,int[] grd,int contact )
    {
          name =nm;
          grade = grd;
          contactHours = contact;
     }
```

Constructor will receive arrays

Tutor class


```
    public void printNames()
    {
        System.out.println("The four tutors are :");
        System.out.println(+name[0]+"\t"+name[1]+"\t"+name[2]+"\t"+name[3]);
    }

    public void printGrades()
    {
        System.out.println(name[0]+"'s grade is : "+ grade[0]);
        System.out.println(name[1]+"'s grade is : "+ grade[1]);
        System.out.println(name[2]+"'s grade is : "+ grade[2]);
        System.out.println(name[3]+"'s grade is : "+ grade[3]);
    }
    public void printContactHours()
    {
        System.out.println("The total contact hours is :"+ contactHours);
    }
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Subject class


```
import java.util.*;

public class Subject
{
    Scanner console = new Scanner(System.in);
    static Tutor myTutor;

    public static void main (String[] args) {
        String[] nameS = new String[4];
        int[] gradeS = new int[4];

        nameS[0] ="cesar";      gradeS[0]=2;
        nameS[1] ="joshua";     gradeS[1]=2;
        nameS[2] ="joe";        gradeS[2]=2;
        nameS[3] ="michael";    gradeS[3]=2;
        int contact = 8;

        myTutor = new Tutor(nameS,gradeS,contact);
        myTutor.printNames();
        myTutor.printGrades();
        myTutor.printContactHours();
```

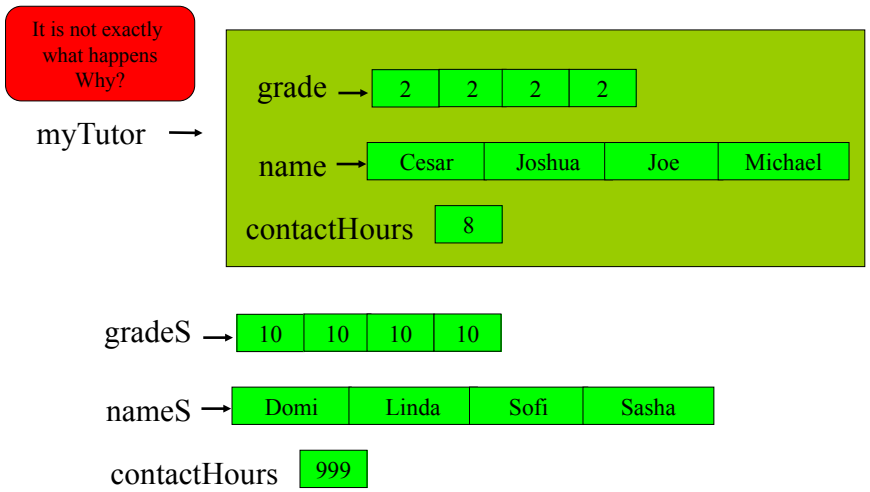

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Subject class


```
    System.out.println("\n\n\n\n==================================================");
    System.out.println("I am changing the name of tutor and \ntheir" +" contact hours");

    nameS[0] ="Domi";  gradeS[0] = 10;
    nameS[1]= "Linda"; gradeS[1] = 10;
    nameS[2]= "Sofi";  gradeS[2] = 10;
    nameS[2]= "Sasha"; gradeS[2] = 10;
    contact = 999;

    System.out.println("\n\n\n\n==================================================");
    System.out.println("Now, lets see what happens after we make these changes");


    System.out.println("\n\n\n\n==========================================================")
    ;
     myTutor.printNames();
     myTutor.printGrades();
     myTutor.printContactHours();
    }
}
```

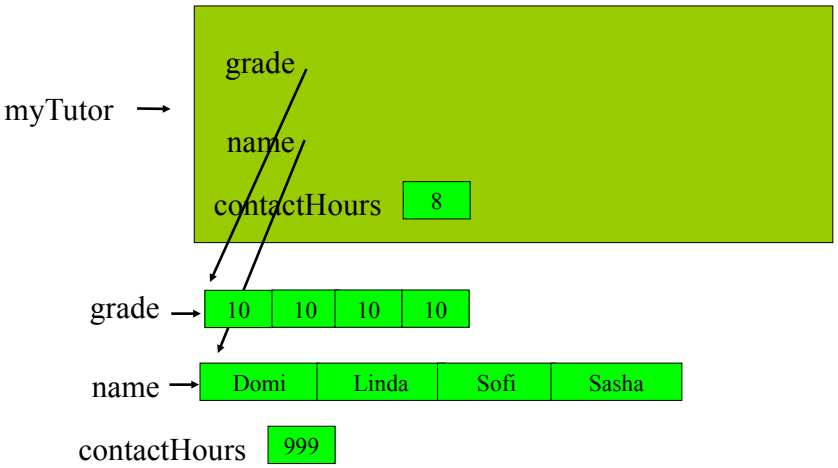

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example Tutor



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example Tutor



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example Tutor

- How to solve this problem…?
- In the constructor of tutor class

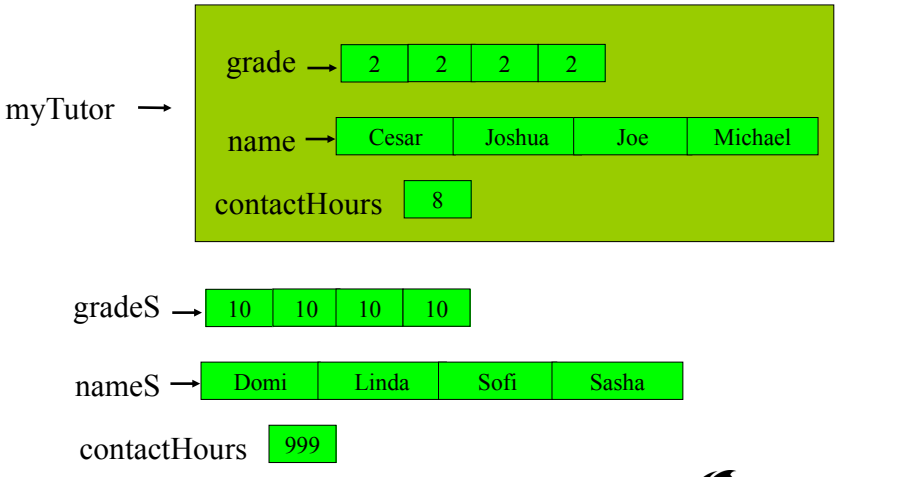
THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Example Tutor


```
public Tutor(String[] nm, int[] grd, int contact )
{
        name = copyName(nm);
        grade = copyGrade(grd);
        contactHours = contact;
}
public String[] copyName(String[] n)
{
        String[] newName = new String[n.length];
        for(int i=0; i<n.length; i++)
           newName[i]=n[i];
        return(newName);
}
public int[] copyGrade(int[] n)
{
        int[] newGrade = new int[n.length];
        for(int i=0; i<n.length; i++)
           newGrade[i]=n[i];
        return(newGrade);
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

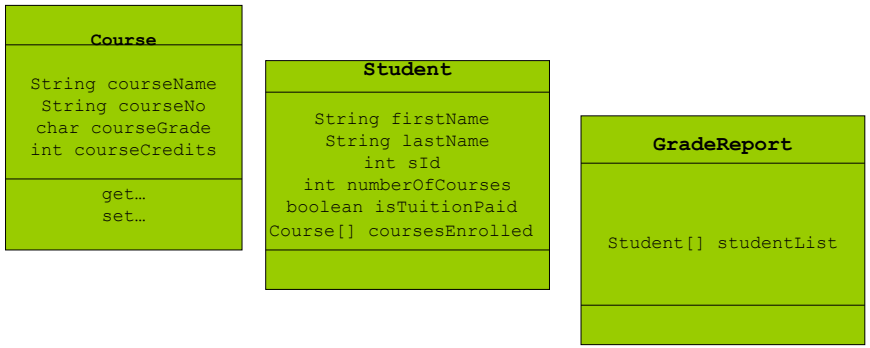
Example Tutor



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

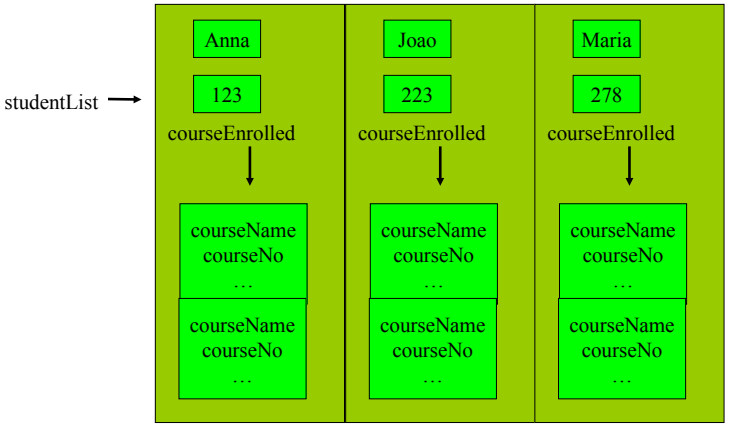
GradeReport example

- Arrays of objects with arrays



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

GradeReport example – the idea



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Class Course


```
public class Course
{
    private String courseName, courseNo;
    private char courseGrade;
    private int courseCredits;

    //Default Constructor
    public Course() {
          courseName = "";  courseNo = "";
          courseGrade = '*';  courseCredits = 0;
    }
    //Constructor
    public Course(String cName, String cNo, char grade, int credits) {
          courseName = cName;  courseNo = cNo;
          courseGrade = grade;  courseCredits = credits;
    }
    public void setCourseInfo(String cName, String cNo, char grade, int credits)  {
          courseName = cName;  courseNo = cNo;
          courseGrade = grade;   courseCredits = credits;
      }
```

What is the difference between Course and setCourseInfo methods?

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Class Course


```
public void setCourseName(String cName) {
     courseName = cName;
  }

public void setCourseNumber(String cNo) {
    courseNo = cNo;
}

public void setCourseGrade(char grade) {
      courseGrade = grade;
}

public void setCourseCredits(int credits) {
      courseCredits = credits;
}
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Class Course


```
public String getCourseName() {
         return courseName;
    }
public String getCourseNumber() {
           return courseNo;
    }
public int getCredits()  {
          return courseCredits;
    }
public char getGrade()  {
           return courseGrade;
    }
public String getCourseInfo(boolean isGrade) {
        String str= courseNo + "\t  " + courseName + "\t\t" + courseCredits + "\t";
        if(isGrade) str = str + courseGrade;
        else        str = str + "***";
        return str;
    }
```


Class Course


```
public void copyCourseInfo(Course otherCourse)
{
        courseName = otherCourse.courseName;
        courseNo = otherCourse.courseNo;
        courseGrade = otherCourse.courseGrade;
        courseCredits = otherCourse.courseCredits;
}
```


Class Student


```
public class Student
{
    private String firstName, lastName;
    private int sId,numberOfCourses;
    private boolean isTuitionPaid;
    private Course [] coursesEnrolled;

    //Default constructor
    public Student(){
        firstName=""; lastName = ""; numberOfCourses = 0; Id = 0; isTuitionPaid = false;
        coursesEnrolled = new Course[6];
        for(int i = 0; i < 6; i++) coursesEnrolled[i] = new Course();
    }

    public void setInfo(String fName, String lName, int ID, int nOfCourses, boolean
    isTPaid, Course[] courses){
         firstName = fName; lastName = lName;
         sId = ID; isTuitionPaid = isTPaid;
         numberOfCourses = nOfCourses ;
         for(int i = 0; i < numberOfCourses; i++)
            coursesEnrolled[i].copyCourseInfo(courses[i]);
    }
```


Class Student


```
public void setName(String first, String last) {
         firstName = first;
         lastName = last;
    }
public void setStudentId(int ID) {
         sId = ID;
     }
public void setIsTuitionPaid(boolean isTPaid) {
         isTuitionPaid = isTPaid;
    }
public void setNumberOfCourses(int nOfCourses) {
         numberOfCourses = nOfCourses ;
    }
public void setCoursesEnrolled(Course[] courses)  {
         for(int i = 0; i < numberOfCourses; i++)
             coursesEnrolled[i].copyCourseInfo(courses[i]);
}
```


Class Student


```
public String getName() {
        return(firstName + " " + lastName);
    }
public int getStudentId() {
         return sId;
     }
public boolean getIsTuitionPaid()   {
         return isTuitionPaid;
     }
public int getNumberOfCourses()  {
         return numberOfCourses;
     }
public Course getCourse(int i)  {
         Course temp = new Course();
         temp.copyCourseInfo(coursesEnrolled[i]);
         return temp;
     }
public int getHoursEnrolled()   {
          int totalCredits = 0;
          for(int i=0; i < numberOfCourses; i++)
              totalCredits += coursesEnrolled[i].getCredits();
          return totalCredits;
Apr-17 }
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Class Student


```
public double getGpa()  {
    double sum = 0.0;
      for(int i=0; i < numberOfCourses; i++) {
           switch(coursesEnrolled[i].getGrade()) {
                   case 'A': sum += coursesEnrolled[i].getCredits() * 4; break;
                   case 'B': sum += coursesEnrolled[i].getCredits() * 3; break;
                   case 'C': sum += coursesEnrolled[i].getCredits() * 2; break;
                   case 'D': sum += coursesEnrolled[i].getCredits() * 1; break;
                   case 'F': sum += coursesEnrolled[i].getCredits() * 0; break;
                   default: System.out.println("Invalid Course Grade");
           }
       }
      return sum / getHoursEnrolled();
    }
public double billingAmount(double tuitionRate) {
         return tuitionRate * getHoursEnrolled();
    }
```

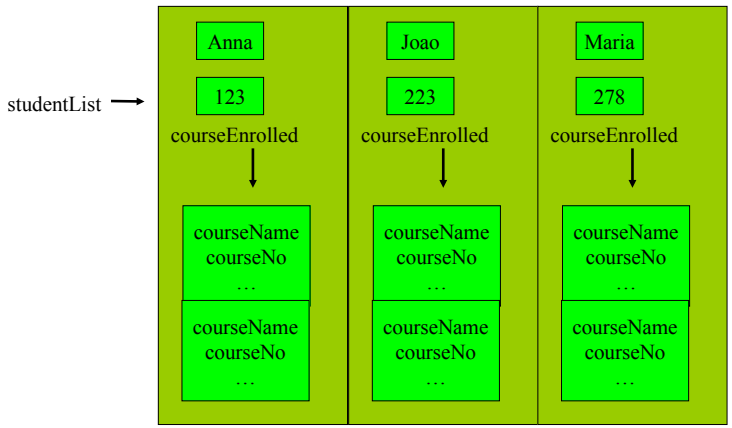

THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Class GradeReport


```
public class GradeReportProgram
{
    static Student[] studentList = new Student[10];
    static int noOfStudents;
    static double tuitionRate;
    static Scanner console = new Scanner(System.in);

    public static void main(String[] args)
    {
          for(int i = 0; i < maxNumberOfStudents; i++)
                    studentList[i] = new Student();

          noOfStudents = console.nextInt();
          tuitionRate = console.nextdouble();

          getStudentData();

          for(int i=0; i<noOfStudents; i++)
                    displayGradeReports(i);
          }
    }
```


THE UNIVERSITY OF NEWCASTLE AUSTRALIA

GradeReport example – the idea



THE UNIVERSITY OF NEWCASTLE AUSTRALIA

Class GradeReport


```
public static void getStudentData()
    String fName, lName, cName, cNo;
    int ID, noOfCourses, credits, count, i;
    char isPaid, grade;
    boolean isTuitionPaid;
    Course[] courses = new Course[6];

    for(i=0; i < 6; i++) courses[i] = new Course();

    for(count=0; count<noOfStudents; count++)
    {
          fname = console.next();
          lname = console.next();
          ID = console.nextInt();
          isPaid = console.next().charAt(0);
          if(isPaid == 'Y') isTuitionPaid = true;
          else              isTuitionPaid = false;

          noOfCourses = console.nextInt();
          for(i = 0; i < noOfCourses; i++) {
                cName = console.next();
                cNo = console.next();
                credits = console.nextInt();
                grade = console.next().charAt();
                courses[i].setCourseInfo(cName, cNo, grade, credits);
             }
          studentList[count].setInfo(fName, lName, ID, noOfCourses, isTuitionPaid, courses);
       } //end for
}
```

Student information

course information

Class GradeReport


```
public static void displayGradeReports(int stNo)
{
    displayedStudentIndex = stNo;
    String CourseListing = "";
    boolean isPaid = studentList[stNo].getIsTuitionPaid();

    System.out.println("name: "+ studentList[stNo].getName());
    System.out.println("ID: "+ studentList[stNo].getStudentId());
    System.out.println("noCourses: "+ studentList[stNo].getNumberOfCourses());
    System.out.println("hours: "+ studentList[stNo].getHoursEnrolled());

    if(isPaid) System.out.println(("GPA: "+String.format("%.2f",studentList[stNo].getGpa())));
    else       System.out.println(""+"****");

    for(int count = 0; count < studentList[stNo].getNumberOfCourses(); count++) {
            if(count == 0) CourseListing =studentList[stNo].getCourse(count).getCourseInfo(isPaid);
            else     CourseListing += "\n" +studentList[stNo].getCourse(count).getCourseInfo(isPaid);
        }

     if(!isPaid)
        CourseListing += "\n"+ "*** Grades are being held for " + "not paying the tuition. ***\n"
                + "Amount Due: $"+String.format("%.2f",studentList[stNo].billingAmount(tuitionRate));

     System.out.println(CourseListing);
    }
}
```


Grade report example

- Grade report example is an example that has
  - Statements
  - Methods
  - Different classes
  - Arrays
  - String
  - You will receive the code in your tutorial
- It is a good example to review and
- It is a very good example for your assignment2