Chapter 6. Reviews

ITSS Java Programming

Lab 1: If else statement

- Use Eclipse to create a project with the name MyGradesProject.
- Create a file Grades.java (described in next slide). Study the use of if/else control structure
- Build and run the program
- Modify the **Grades.java** as following:
  - If the average is greater than 90 (average > 90), display "You worked too hard! Your average is xx.0."
  - If the average is greater than 50 (average > 50) and less than or equal to 90 (average <= 90), display "You did OK! Your average is xx.0."
  - If the average is less than or equal to 50 (average <= 50), display "You need to do some work! Your average is xx.0."

Lab 1: If else statement

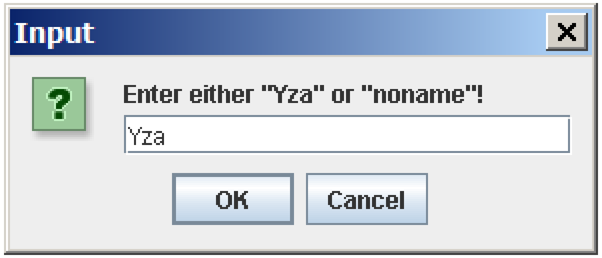
```
import javax.swing.JOptionPane;
public class Grades {
    public static void main(String[] args) {
        int mathGrade = 0;
        int historyGrade = 0;
        int scienceGrade = 0;
        double average = 0;
        mathGrade = Integer.parseInt(JOptionPane.showInputDialog("Enter math grade
              between 0 and 100!"));
        historyGrade = Integer.parseInt(JOptionPane.showInputDialog("Enter history
              grade between 0 and 100!"));
        scienceGrade = Integer.parseInt(JOptionPane.showInputDialog("Enter science
              grade between 0 and 100!"));
```

Lab 1: If else statement

```
// Compute average
        average = (mathGrade+historyGrade+scienceGrade)/3;
        // Perform if & else control
        if (average >= 60){
            JOptionPane.showMessageDialog(null, "Good job! Your average is " +
              average);
        } else{
            JOptionPane.showMessageDialog(null, "You need to do better! Your average
              is " + average);
        }
      }
}
```

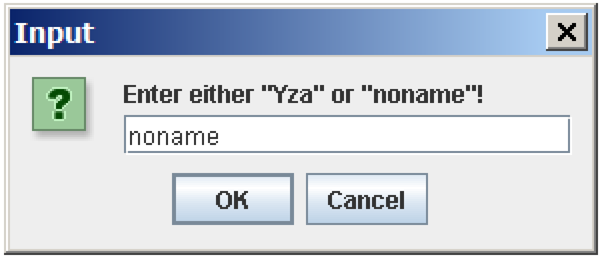
Lab2: For Loop

- Study Forloop.java and complet this program so that it can search the name inputed by user in the array.
- Some image of the execution of program



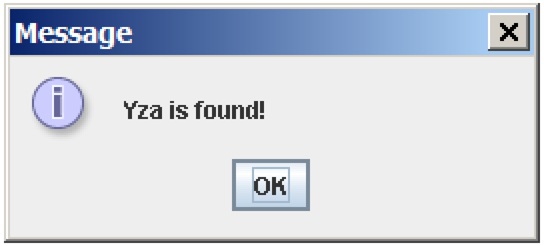
Enter either "Yza" or "noname"!

Yza

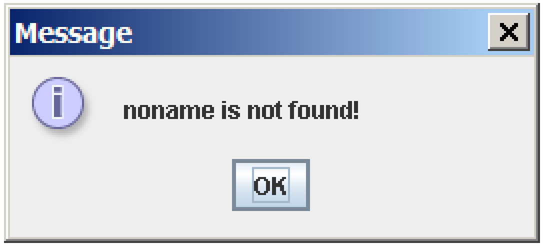


Enter either "Yza" or "noname"!

noname



Yza is found!



noname is not found!

Lab2: For Loop

```
public class ForLoop {

 /** Creates a new instance of ForLoop */
 public ForLoop() {
 }

 /**
  * @param args the command line arguments
  */
 public static void main(String[] args) {
     // Declare and initialize String array variable called names.
     String names []={"Beah","Bianca","Lance","Belle","Nico","Yza","Gem","Ethan"};

     // This is the search string we are going to use to search the array.
     String searchName = JOptionPane.showInputDialog("Enter either \"Yza\" or
      \"noname\"!");

     // Declare and initialize boolean primitive type variable called foundName.
     boolean foundName =false;
```

Lab2: For Loop

```
 // Search the String array using for loop.
//  * The "names.length" is the size of the array.
//  * This for loop compares the value of each entry of the array with
//     the value of searchString String type variable.
//  * The equals(..) is a method of String class. Think about why you
//     cannot use "names[i] == searchName" as comparison logic here.

for (int i=0; i<names.length; i++){
    if (names [i ].equals(searchName)){
        foundName =true;
        break;
    }
}

// Display the result
if (foundName)
    JOptionPane.showMessageDialog(null, searchName + " is found!");
else
    JOptionPane.showMessageDialog(null, searchName + " is not found!");

    }

}
```

Lab3: ArrayList

- Modify program in Lab2 to use an arraylist instead of the array.
- When a name entered by user is not found, add it to the name list.

Lab4: Command Arguments

- The homework is to create a new Eclipse project called "MyOwnCLA" as following:
- Receive the ages of the all your family members (between 3 to 6 members) as command line arguments in the format as following (name then age)
  - Hoang 12 Trang 34 Lan 23
- Compute and display the average of the ages that were entered.

Lab5: Class and Object

- Using Eclipse to create a project named MyStudentRecordProject
- Creat a new class StudentRecord (source code given in next slides)
- Creat a new class StudentRecord and fill in the code of main function that follows the guide in the comments

StudentRecord.java

```
public class StudentRecord {

        /** Creates a new instance of
    StudentRecord */
        public StudentRecord() {
        }

        // Declare instance variables.
        private String name;
        private double mathGrade;
        private double englishGrade;
        private double scienceGrade;


        // Declare static variables.
        private static int studentCount = 0;

        /**
         * Returns the name of the student
         */
        public String getName(){
            return name;
        }

        /**
         * Changes the name of the student
         */
        public void setName(String temp ){
            name =temp;
        }
```

```
/**
    Computes the average of the english, math and
    science grades */
        public double getAverage(){
            double result =0;
            result=(getMathGrade()+getEnglishGrade()
              +getScienceGrade() )/3;
            return result;
        }
/** * Returns the number of instances of
    StudentRecords
         */
        public static int getStudentCount(){
            return studentCount;
        }

        /**
         * Returns the number of instances of
    StudentRecords.
         * This is a static method.
         */
        public static void increaseStudentCount(){
            studentCount++;
        }

        // Instance methods
        public double getMathGrade() {
            return mathGrade;
        }
```

StudentRecord.java

```
    public void setMathGrade(double mathGrade) {
        this.mathGrade = mathGrade;
    }

    public double getEnglishGrade() {
        return englishGrade;
    }

    public void setEnglishGrade(double englishGrade) {
        this.englishGrade = englishGrade;
    }

    public double getScienceGrade() {
        return scienceGrade;
    }

    public void setScienceGrade(double scienceGrade) {
        this.scienceGrade = scienceGrade;
    }
}
```

StudentRecordExample.java

```
public class StudentRecordExample {
    public static void main(String[] args) {
        // Create an object instance of StudentRecord class.
        ...
        // Increament the studentCount by invoking a static method.
        ...

        // Create another object instance of StudentRecord class.
        ...
        // Increament the studentCount by invoking a static method.
        ...

        // Create the 3rd object instance of StudentRecord class.
        ...
        // Increament the studentCount by invoking a static method.
        ...
      // Set the names of the students.
        ... // To Anna
       ... //  To Beah;
        ... // To Cris;

        // Print anna's name.

        // Print number of students.
        ...// ;
    }

}
```

Lab6: Overloading

- Add two overloaded myprint() methods to the StudentRecored.java as shown in the code below. Build and Run project.
-

```
public class StudentRecord {
     :
:
       // Overloaded myprint(..) methods
        public void myprint(){
           System.out.println("First overloaded method: Nothing is passed            on");
        }

        public void myprint(String name ){
           System.out.println("Second overloaded method: Name:"+name);
        }

        public void myprint(String name, double averageGrade){
           System.out.print("Third overloaded method: Name:"+name+" ");
           System.out.println("Average Grade:"+averageGrade);
        }

}
```

Overloading

- Modify StudentRecordExample.java as shown in Code below. Build and Run project.

```
public class StudentRecordExample {

        public static void main(String[] args) {

            // Create an object instance of StudentRecord class.
            ……….

            // Set Anna's grades
            annaRecord.setName("Anna");
            annaRecord.setEnglishGrade(95.5);
            annaRecord.setScienceGrade(100);

            // Invoke overloaded methods
            annaRecord.myprint();
            annaRecord.myprint(annaRecord.getName());
            annaRecord.myprint(annaRecord.getName(), annaRecord.getAverage());

        }

    }
```

Lab7: Constructors

- Add to StudentRecord class the constructor as the specification below:

```
 // Default constructor
 public StudentRecord() {
 }
// Constructor that gets single parameter
 public StudentRecord(String name){
   …..
 }
 // Constructor that gets two parameters
 public StudentRecord(String name, double mGrade){
     …..
 }
 // Constructor that gets three parameters
 public StudentRecord(String name, double mGrade, double eGrade){
     …..
 }
 // Constructor that gets four parameters
 public StudentRecord(String name, double mGrade, double eGrade,
         double sGrade){
     …..
 }
```

StudentRecordExample.java

```
public class StudentRecordExample {

    public static void main(String[] args) {

        // Create an object instance of StudentRecord class.
        StudentRecord annaRecord = new StudentRecord("Anna");

        // Increament the studentCount by invoking a static method.
        StudentRecord.increaseStudentCount();

        // Create another object instance of StudentRecord class.
        StudentRecord beahRecord =new StudentRecord("Beah", 45);

        // Increament the studentCount by invoking a static method.
        StudentRecord.increaseStudentCount();

        // Create the 3rd object instance of StudentRecord class.
        StudentRecord crisRecord =new StudentRecord("Cris", 23.3, 67.45, 56);

        // Increament the studentCount by invoking a static method.
        StudentRecord.increaseStudentCount();

        // Print Cris' name and average
        System.out.println("Name = " + crisRecord.getName() + " Average = " +
          crisRecord.getAverage());

        // Print number of students.
        System.out.println("Student Count = "+StudentRecord.getStudentCount());
    }

}
```

Exercise

- The **Student** class has **StudentRecord** class as an instance variable. Name it as **studentRecord**. You can use the StudentRecord class from the MyStudentRecordProject above or you can create a new one - the only requirement is that it has to have at least one instance variable of its own.

- The **Student** class has **studentId** instance variable whose type is **Integer** type.
- Move the **studentCount** static variable from the **StudentRecord** class to **Student** class.

- Rewrite main.java as following
  - Create 3 instances of Student class and initialize them accordingly - use whatever initialization values that are appropriate.

  - Display the information of each student including the student id, name.
  - Display the studentCount.

Lab8: Inheritance

- Create a new project name MyOnlineShop
- Create a class Product in Product.java(code given)

```
package myonlineshop;

    public class Product {

        private double regularPrice;

        /** Creates a new instance of Product */
        public Product(double regularPrice) {
            this.regularPrice = regularPrice;
        }

        // Method that will be overridden
        public double computeSalePrice(){
            return 0;
        }

        public double getRegularPrice() {
            return regularPrice;
        }

        public void setRegularPrice(double regularPrice) {
            this.regularPrice = regularPrice;
        }

    }
```

Lab8: Inheritance

- Write Electronics.java that extends Product class.

```
package myonlineshop;

    public class Electronics extends Product{

       private String manufacturer;

       /** Creates a new instance of Electronics */
       public Electronics(double regularPrice,
             String manufacturer) {
          super(regularPrice);
          this.manufacturer = manufacturer;
       }
       // Override this method
       public double computeSalePrice(){
          return super.getRegularPrice() * 0.6;
       }
       public String getManufacturer() {
          return manufacturer;
       }
       public void setManufacturer(String manufacturer) {
          this.manufacturer = manufacturer;
       }

    }
```

Lab8: Inheritance

- Write MP3Player.java that extends Electronics class.

```
package myonlineshop;

    MP3Player extends Electronics {

        private String manufacturer;

        /** Creates a new instance of Electronics */
        public MP3Player(double regularPrice,
                    String manufacturer,
                    String color) {
        ....;
        }
        // Override this method
        public double computeSalePrice(){
            // Price is calculated by the rate of 0.9 of the RegularPrice
        }
        .........
    }
```

Lab8: Inheritance

- Write TV.java that extends Electronics class as the code given below.

```
package myonlineshop;

    public class TV extends Electronics {

        int size;

        /** Creates a new instance of TV */
        public TV(double regularPrice,
                String manufacturer,
                int size) {
            super(regularPrice, manufacturer);
            this.size = size;
        }

        // Override this method
        public double computeSalePrice(){
            return super.getRegularPrice() * 0.8;
        }
    }
```

Lab8: Inheritance

Create the Book class that extends Product class as the code below

```
package myonlineshop;

    public class Book extends Product{

        private String publisher;
        private int yearPublished;

        /** Creates a new instance of Book */
        public Book(double regularPrice,
                String publisher,
                int yearPublished) {
            super(regularPrice);
            this.publisher = publisher;
            this.yearPublished = yearPublished;
        }

        // Override this method
        public double computeSalePrice(){
            return super.getRegularPrice() * 0.5;
        }

        public String getPublisher() {
            return publisher;
        }

        public void setPublisher(String publisher) {
            this.publisher = publisher;
        }

        public int getYearPublished() {
            return yearPublished;
        }

        public void setYearPublished(int
    yearPublished) {
            this.yearPublished = yearPublished;
        }

    }
```

Lab8: Inheritance

- Write a Main Class that create and store objects in an array of five Products. The objects are:
  - a TV: Price = 1000, Manufacturer: “Samsung”, size:30;
  - a TV: Price = 2000, Manufacturer: “Sony”, size:50;
  - a MP3Player: Price = 250, Manufacturer: “Apple”, color:blue;
  - a Book: Price = 34, Publisher: “Sun press”, Year=1992
  - a Book: Price = 15, Publisher: “HUT Press”, Year=1986
- Print out the total regular price and total sale price and the detail regular price and sale price for each product.

Exercise

- Add **Camera** class which is a sub-class of **Eletronics** class
  - Compute the sale price of Camera with the following business logic
  - Regular price * 0.7
  - In the **Main.java**, initialize two object instances of Camera class

Lab 9: Polymorphism – Abstract class

- Modify the Product class so that it becomes an abstract class with the abstract method computeSalePrice
- Write two new class extends class Book:
  - ChildrenBook which has an additional Attribute: age (int) and rate for Sale Price is 0.3
  - Cartoon which has an additional Attribute: characterName (String) and rate for Sale Price is 0.4
- Modify Main.java by adding 2 objects of Cartoon and ChildrenBook class as follows:
  - ChildrenBook: RegPrice=15, Publisher= “Kim Dong press", Year= 1987, Age= 8
  - Cartoon: RegPrice=14, Publisher= “Marvel press", 1924, "Batman");
- Run and build the project

Solution: Product.java

```
package myonlineshop;

public abstract class Product {

    private double regularPrice;

    /** Creates a new instance of Product */
    public Product(double regularPrice) {
        this.regularPrice = regularPrice;
    }

    // Abstract method
    public abstract double computeSalePrice();

    public double getRegularPrice() {
        return regularPrice;
    }

    public void setRegularPrice(double regularPrice) {
        this.regularPrice = regularPrice;
    }

}
```

Solution: ChildrenBook.java

```
package myonlineshop;

    public class ChildrenBook extends Book{

        int age; // age this book is written for

        /** Creates a new instance of ChildrenBook */
        public ChildrenBook(double regularPrice,
                String publisher,
                int yearPublished,
                int age) {
            super(100, "Sun press", 2002);
            this.age = age;
        }

        // Override this method
        public double computeSalePrice(){
            return super.getRegularPrice() * 0.3;
        }

    }
```

Solution: Cartoon.java

```
package myonlineshop;

    public class Cartoon extends Book {

        String characterName;

        /** Creates a new instance of Cartoon */
        public Cartoon(double regularPrice,
                String publisher,
                int yearPublished,
                String characterName) {
            super(150, "Sun press", 1978);
            this.characterName = characterName;
        }

        // Override this method
        public double computeSalePrice(){
            return super.getRegularPrice() * 0.4;
        }

    }
```

Main.java

```
// Create object instances and assign them to
        // the type of Product.
        pa[0] = new TV(1000, "Samsung", 30);
        pa[1] = new TV(2000, "Sony", 50);
        pa[2] = new MP3Player(250, "Apple", "blue");
        pa[3] = new Book(34, "Sun press", 1992);
        pa[4] = new Book(15, "Korea press", 1986);
        pa[5] = new ChildrenBook(15, “Kim Dong
  press", 1987, 8);
        pa[6] = new Cartoon(14, “Marvel press", 1924,
  "Batman");
```

Lab10: Interface

- Modify the project MyOnlineShop in previous lab by using interfaces.
- Declare ProductInterface (code given)
- Declare ElectronicsInterface (code given)
- Modify the classes Product, Electronics to implement these interface
- Write BookInterface.java

Lab10: Interface

- Create an interface BookInterface as follows:

```
package myonlineshopusinginterface;

  public interface BookInterface extends ProductInterface {
     public String getPublisher();
     public void setPublisher(String publisher);
     public int getYearPublished();
     public void setYearPublished(int yearPublished);
  }
```

- Modify the Book class to implement this interface.
- Build and Run the project.

Code given

```
package myonlineshopusinginterface;
    public interface ElectronicsInterface extends ProductInterface {
     public String getManufacturer();
   }

package myonlineshopusinginterface;

    public interface ProductInterface {
           public double computeSalePrice();
           public double getRegularPrice();
           public void setRegularPrice(double regularPrice);
    }
```