10. Express the result into a GUI frame.
11. Check the Java file for errors and right-click the class in the project to display the context menu. Choose Run, Java Application in the context menu to run the class.
12. Save screenshots of the output similar to Figures 8-2-3 and 8-2-4 and submit them to your instructor.

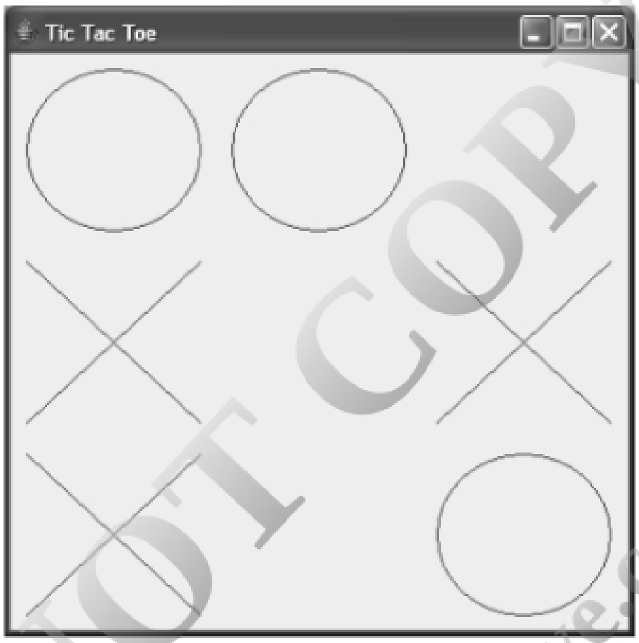


Figure 8-2-3

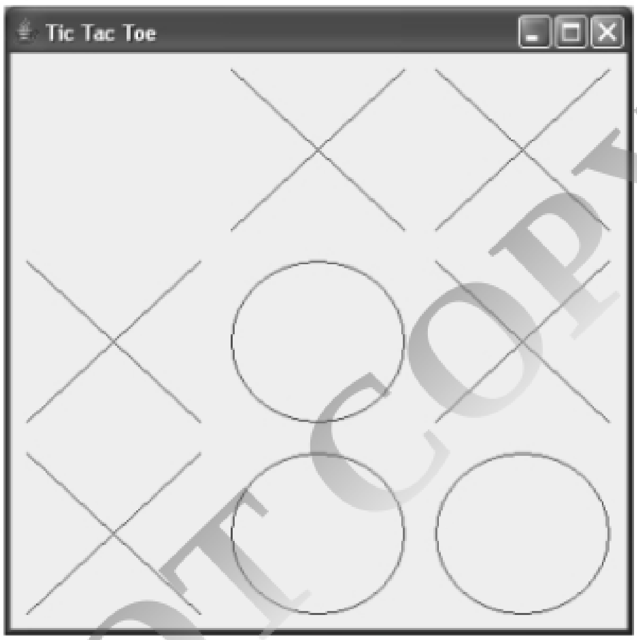


Figure 8-2-4

Did it work?

- Were you able to display a frame with nine random icon images (X, O, or blank) using the Graphics class?

Summary and Reminders

Summary

This unit examined how to get started with GUI programming by using component class. In addition, it explained how to draw things by using the Graphics class. The next unit will look at event-driven programming and more GUI programming in Java applications.

Reminders

- Students should review Chapter 12 and 13 of Introduction to Java Programming before starting Lab 7.1 and 7.2.

(End of Unit 7)

Lab 8-1: Displaying Calendars Using the GregorianCalendar Class

Purpose

In Lab 4.2, you wrote code to print a calendar. In this lab exercise, you will rewrite the PrintCalendar class from Lab 4.2 to display a calendar for a specified month using the Calendar and GregorianCalendar classes. The program receives the month and year from the command line, for example, to display a calendar for August 2013.

How it Fits In

Students will continue working with the basic building blocks of Java programs. They will work with a Java statement that will deal with subclasses and the GregorianCalendar.

Objectives

- Create a Java program
- Create a sub class
- Create a GUI calendar

Estimated Completion Time: 50 minutes

Task 1: Study Material

Read Chapter 14 from *Introduction to Java Programming*, Brief Version, (9th ed) by Liang, Y. D. (2013).

- Page references
  - Page 560, Section 15.2 (Abstract Classes)
  - Page 567, Section 15.4 (Case Study: Calendar and GregorianCalendar)
  - Page 568, Section 15.6 (TestCalendar.java)

Eclipse and Java websites for additional reference

- http://www.cs.armstrong.edu/liang/intro9e/supplement.html
- http://help.eclipse.org/juno/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-cheatsheets.htm

Keywords

- Subclass
- PrintMonth
- GregorianCalendar

Task 2: Create a Java Program to Display a Calendar

Procedure

1. Open the Eclipse software that is installed on the Lab computer.
2. Create a subclass of GregorianCalendar called MyCalendar.
3. Create the following methods:
   a. public int daysInMonth()
   b. public String getMonthName()

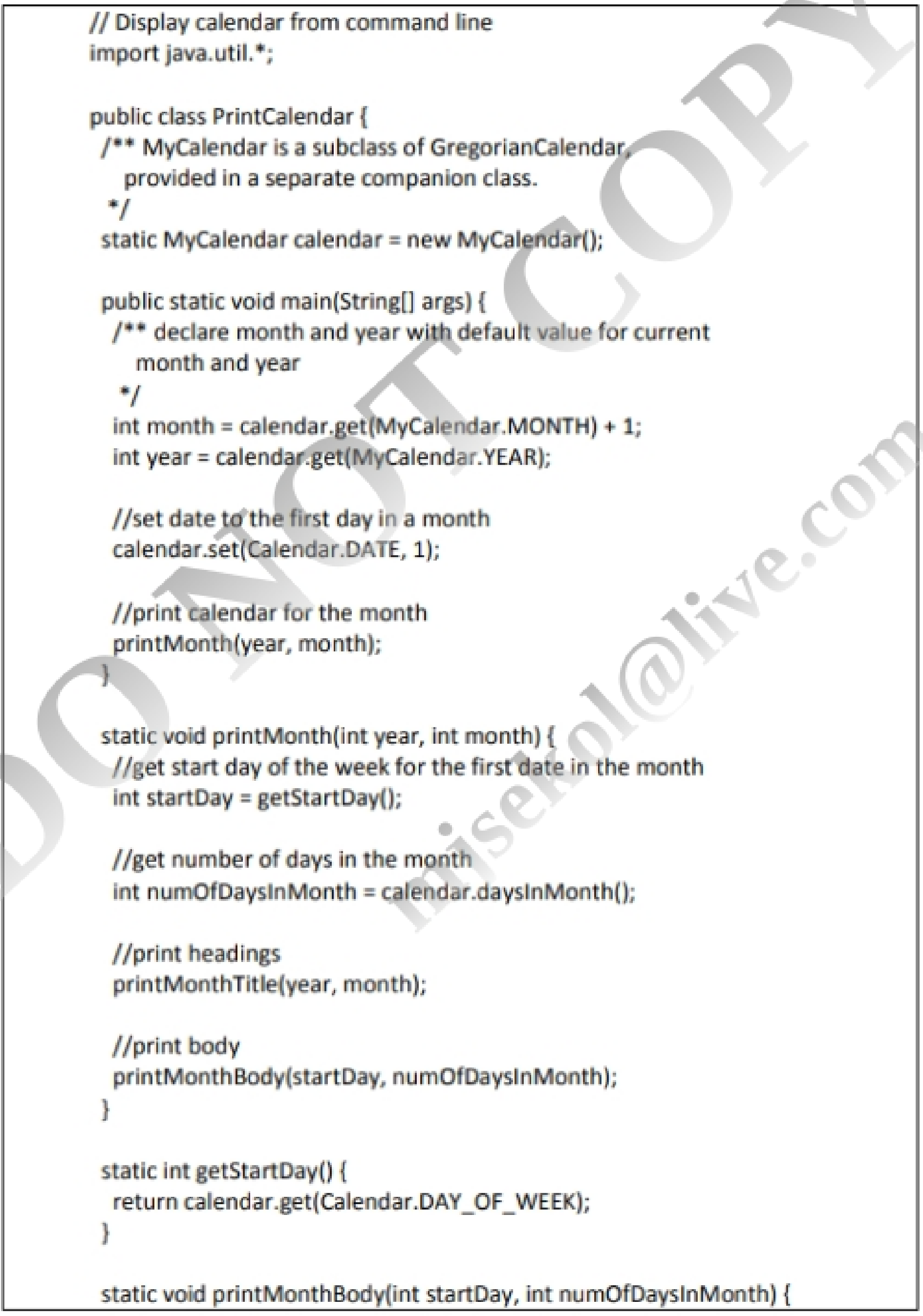
OR, complete the code shown in Figure 8-1-1.

```
// MyCalendar.java: A subclass of GregorianCalendar
import java.util.*;

public class MyCalendar extends GregorianCalendar {
  /**Find the number of days in a month*/
  public int daysInMonth() {

  }

  /**return month name in English*/
  public String getMonthName() {

  }
}
```

Figure 8-1-1

4. Create a java class called PrintCalendar to display calendar from command line.
5. Create the following methods:
   a. static void printMonth(int year, int month)
   b. static int getStartDay()
   c. static void printMonthBody(int startDay, int numOfDaysInMonth)

OR, complete the code picture in Figure 8-1-2 (below).



```
// Display calendar from command line
import java.util.*;

public class PrintCalendar {
 /** MyCalendar is a subclass of GregorianCalendar,
    provided in a separate companion class.
  */
 static MyCalendar calendar = new MyCalendar();

 public static void main(String[] args) {
  /** declare month and year with default value for current
     month and year
   */
  int month = calendar.get(MyCalendar.MONTH) + 1;
  int year = calendar.get(MyCalendar.YEAR);

  //set date to the first day in a month
  calendar.set(Calendar.DATE, 1);

  //print calendar for the month
  printMonth(year, month);
 }

 static void printMonth(int year, int month) {
  //get start day of the week for the first date in the month
  int startDay = getStartDay();

  //get number of days in the month
  int numOfDaysInMonth = calendar.daysInMonth();

  //print headings
  printMonthTitle(year, month);

  //print body
  printMonthBody(startDay, numOfDaysInMonth);
 }

 static int getStartDay() {
  return calendar.get(Calendar.DAY_OF_WEEK);
 }

 static void printMonthBody(int startDay, int numOfDaysInMonth) {
        //print padding space before the first day of the month

      }
    }
```

Figure 8-1-2

6. Check the Java file for errors and right-click the class in the project to display the context menu. Choose Run, Java Application in the context menu to run the class.
7. Save screenshots of the output and submit them to your instructor.

Did it work?

Were you able to:

- Display current monthly calendar by the PrintCalendar class with parameters of current month and current year.
- Display current monthly calendar by the PrintCalendar class with parameter of current month.
- Display current monthly calendar by the PrintCalendar class without any parameters.

Lab 8-2: The Person and Student Classes

Purpose

Design the Student class that extends Person. Implement the compareTo method in the Person class to compare persons in alphabetical order of their last name. Implement the compareTo method to compare students in alphabetical order of their major and last name.

Write a test program with the following four methods:

```
/** Sort an array of comparable objects **/
public static void sort(Object[] list)
/** Print an array of objects **/
public static void printList(Object[] object)
/** Return the max object in an array of comparable objects **/
public static Object max(Object[] list)
```

Main method: Test the sort, printList, and max methods using an array of four students, an array of four strings, an array of 100 random rationals, and an array of 100 random integers.

How it Fits In

Students will continue working with the basic building blocks of Java programs. They will work with a Java statement that will deal with sorting a list and the arrays of comparable objects.

Objectives

- Create a Java program
- Create a list
- Create an array

Estimated Completion Time: 40 minutes

Task 1: Study Material

Read Chapter 14 from *Introduction to Java Programming*, Brief Version, (9th ed) by Liang, Y. D. (2013).

- Page references
  - Page 267, Section 7.3 (Processing Two-Dimensional Arrays
  - Page 429, Section 11.10 (The Objects Equal Method)

- Page 336, Section 9.2 (The String Class)

Eclipse and Java websites for additional reference

- http://www.cs.armstrong.edu/liang/intro9e/supplement.html
- http://help.eclipse.org/juno/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-cheatsheets.htm

Keywords

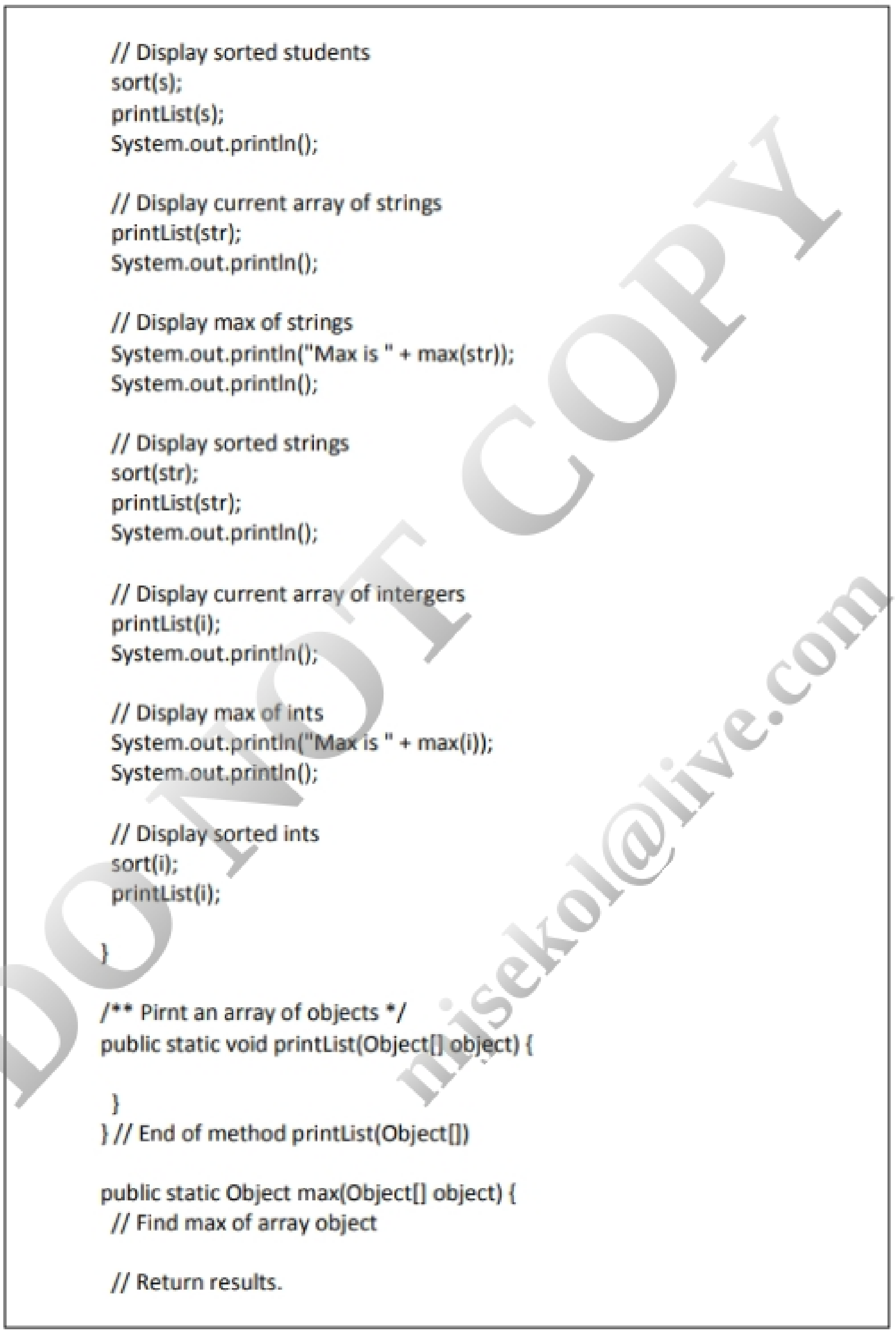
- Array
- Sort
- String

Task 2: Create a Java Program to Display Person and Student Classes

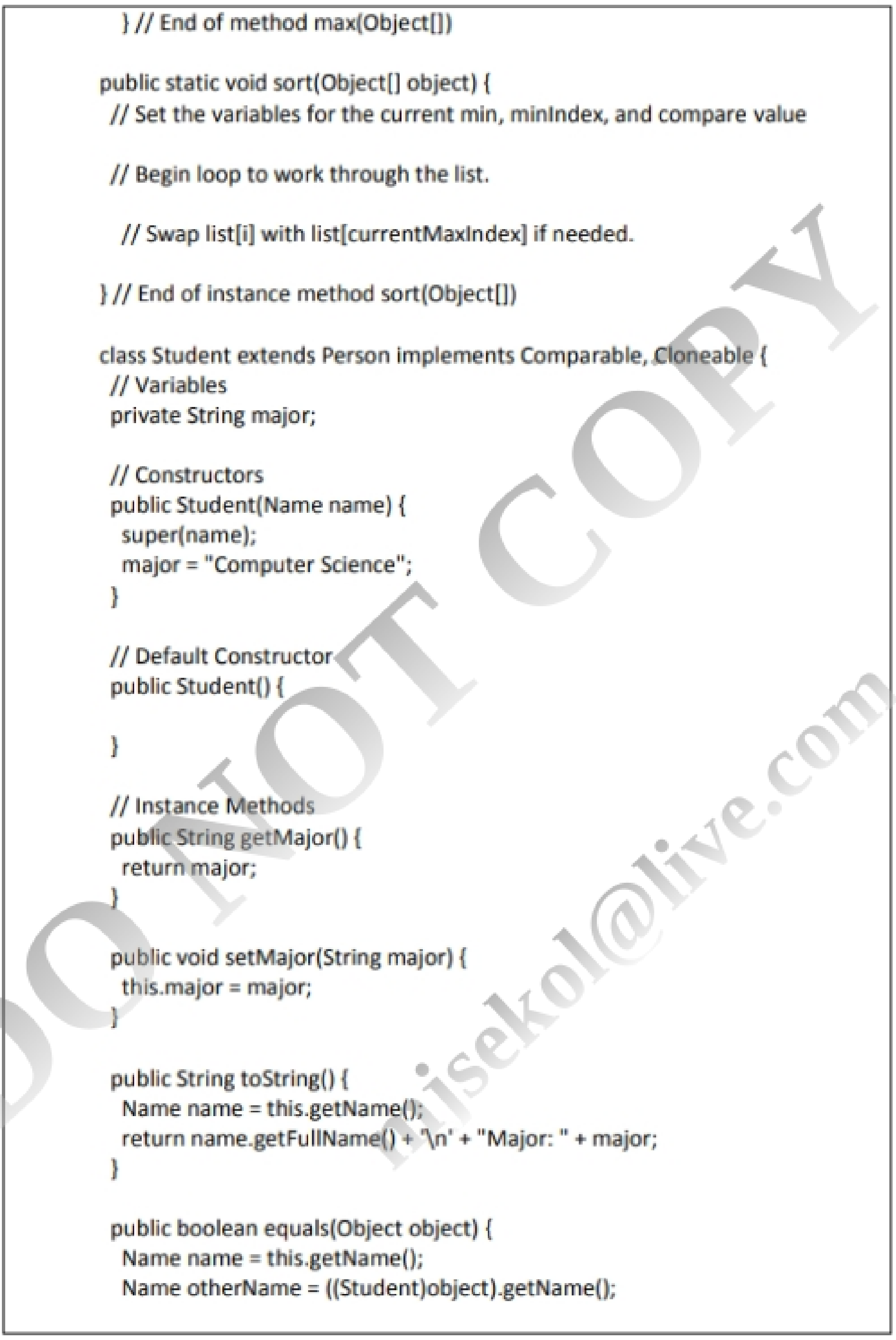
Procedure

1. Open the Eclipse software that is installed on the Lab computer.
2. Create a Java class called School.
3. Using the sample code provided complete the code shown in Figure 8-2-1.

```
public static void main(String[] args) {
  new School();
}

public School() {
  // Define the arrays to be tested.
  Student[] s = new Student[3];
  s[0] = new Student(new Name("Derek", 'S', "Dexony"));
  s[1] = new Student(new Name("Stacy", 'M', "Waters"));
  s[2] = new Student(new Name("Adamo", 'U', "Leetmz"));

  String[] str = {"Orange", "Jackblade", "Apple"};
  Integer[] i = {new Integer(5), new Integer(0), new Integer(3)};

  // Display current array of students
  printList(s);
  System.out.println();

  // Display max of students
  System.out.println("Max is " + max(s));
  System.out.println();
```



```
    // Display sorted students
    sort(s);
    printList(s);
    System.out.println();

    // Display current array of strings
    printList(str);
    System.out.println();

    // Display max of strings
    System.out.println("Max is " + max(str));
    System.out.println();

    // Display sorted strings
    sort(str);
    printList(str);
    System.out.println();

    // Display current array of intergers
    printList(i);
    System.out.println();

    // Display max of ints
    System.out.println("Max is " + max(i));
    System.out.println();

    // Display sorted ints
    sort(i);
    printList(i);

   }

   /** Pirnt an array of objects */
   public static void printList(Object[] object) {

    }
   } // End of method printList(Object[])

   public static Object max(Object[] object) {
    // Find max of array object

    // Return results.
```



```
    } // End of method max(Object[])

  public static void sort(Object[] object) {
    // Set the variables for the current min, minIndex, and compare value

    // Begin loop to work through the list.

      // Swap list[i] with list[currentMaxIndex] if needed.

  } // End of instance method sort(Object[])

  class Student extends Person implements Comparable, Cloneable {
    // Variables
    private String major;

    // Constructors
    public Student(Name name) {
      super(name);
      major = "Computer Science";
    }

    // Default Constructor
    public Student() {

    }

    // Instance Methods
    public String getMajor() {
      return major;
    }

    public void setMajor(String major) {
      this.major = major;
    }

    public String toString() {
      Name name = this.getName();
      return name.getFullName() + '\n' + "Major: " + major;
    }

    public boolean equals(Object object) {
      Name name = this.getName();
      Name otherName = ((Student)object).getName();
```