CSC 236 - Lab 1 (4 programs) Review of CSC 162

From the Gaddis (4th ed.) textbook

1. P. 600 #8 (classes and objects)
Create the following objects:

ParkedCar: Volkswagon, 1972, Red, license no. 147RHZM, 125 mins. Parked
ParkingMeter: 60 mins.
PoliceOfficer: Joe Friday, badge no. 4788

Output:
Show the Car Data and Officer Data

Did the officer issue a ticket?
If so, print the no. of mins. illegally parked and the fine in dollars.
Else, print no crimes committed

Run the program a second time using 60 mins. of parking with the same data

You should have the following classes and methods:

- ParkingTicketSimulator - has the main( ) program
- ParkedCar class - show the constructor, copy constructor, setMake, setModel, setColor, setLicenseNumber, setMinutesParked, getMake, getModel, getColor, getLicenseNumber, getMinutesParked, toString
- ParkingMeter class - constructor, setMinutesPurchased, getMinutesPurchased
- ParkingTicket class - constructor, copy constructor, calculateFine, setCar, setOfficer, getCar, getFine, getOfficer, toString
- PoliceOfficer class - constructor, copy constructor, setName, setBadgeNumber, getName, getBadgeNumber, patrol (looks at no. of mins parked and mins. purchased), toString

2. P. 733 #1 (inheritance/polymorphism)

WorkerDemo class has main( ): Create a ProductionWorker object passing the data to the constructor:

John Smith, Employee no. 123-A, Hire: 11-15-2005, Shift: Day, Hourly rate: $16.50

Create another ProductionWorker object and use the set methods. Joan Jones, Employee no. 222-L, Hire: 12-12-2005, Shift: Night, Hourly rate: $18.50

Output these 2 production workers.

Show the following methods in these classes:

Employee class - default constructor, overloaded constructor, setName, setEmployeeNumber, setHireDate, getName, getEmployeeNumber, getHireDate, isValidEmpNum, toString

ProductionWorker class - default constructor, overloaded constructor, setShift, setPayRate, getShift, getPayRate, toString (format $)

3. P. 792 #10 (exceptions)

WorkerDemo class has main( ):

a) Create a ProductionWorker object with valid data using a createWorker method

John Smith, Employee no. 123, Hire: 11-15-2009, Shift: Day, Hourly rate: $16.50

b) Use an invalid employee number

John Smith, Employee no. 10001, Hire: 11-15-2009, Shift: Day, Hourly rate: $16.50

c) Use an invalid shift number

John Smith, Employee no. 123, Hire: 11-15-2009, Shift: 66, Hourly rate: $16.50

d) Use a negative pay rate

John Smith, Employee no. 123, Hire: 11-15-2009, Shift: Day, Hourly rate: $-99.00

The createWorker method should have try-catch blocks

Show the output for the above 4 cases.

Also, show the following classes:

Employee class - modify the class for exceptions
ProductionWorker class - modify the class for exceptions
InvalidEmployeeNumber class, InvalidPayRate class, InvalidShift class

4. P. 960 #2

RecursiveIsMember class - has main( )

Initialize a 1-D array with the values: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20

Using recursion, test all values from 0 to 20 and see if they are contained in the array. Print the results out for all numbers 0-20.

 CSC 236 - Lab 1

- A `greaterThan` method that accepts a `Month` object as an argument. If the calling object's `monthNumber` field is greater than the argument's `monthNumber` field, this method should return `true`. Otherwise, it should return `false`.
- A `lessThan` method that accepts a `Month` object as an argument. If the calling object's `monthNumber` field is less than the argument's `monthNumber` field, this method should return `true`. Otherwise, it should return `false`.

6. `CashRegister` Class

Write a `CashRegister` class that can be used with the `RetailItem` class that you wrote in Chapter 6's Programming Challenge 4. The `CashRegister` class should simulate the sale of a retail item. It should have a constructor that accepts a `RetailItem` object as an argument. The constructor should also accept an integer that represents the quantity of items being purchased. In addition, the class should have the following methods:

- The `getSubtotal` method should return the subtotal of the sale, which is the quantity multiplied by the price. This method must get the price from the `RetailItem` object that was passed as an argument to the constructor.
- The `getTax` method should return the amount of sales tax on the purchase. The sales tax rate is 6 percent of a retail sale.
- The `getTotal` method should return the total of the sale, which is the subtotal plus the sales tax.

Demonstrate the class in a program that asks the user for the quantity of items being purchased, and then displays the sale's subtotal, amount of sales tax, and total.

7. Sales Receipt File

Modify the program you wrote in Programming Challenge 6 to create a file containing a sales receipt. The program should ask the user for the quantity of items being purchased, and then generate a file with contents similar to the following:

```
SALES RECEIPT
Unit Price: $10.00
Quantity: 5
Subtotal: $50.00
Sales Tax: $ 3.00
Total: $53.00
```

8. Parking Ticket Simulator

For this assignment you will design a set of classes that work together to simulate a police officer issuing a parking ticket. You should design the following classes:

- **The `ParkedCar` Class:** This class should simulate a parked car. The class's responsibilities are as follows:
  - To know the car's make, model, color, license number, and the number of minutes that the car has been parked.
- **The `ParkingMeter` Class:** This class should simulate a parking meter. The class's only responsibility is as follows:
  - To know the number of minutes of parking time that has been purchased.

- The `ParkingTicket` Class: This class should simulate a parking ticket. The class's responsibilities are as follows:
  - To report the make, model, color, and license number of the illegally parked car
  - To report the amount of the fine, which is $25 for the first hour or part of an hour that the car is illegally parked, plus $10 for every additional hour or part of an hour that the car is illegally parked
  - To report the name and badge number of the police officer issuing the ticket
- The `PoliceOfficer` Class: This class should simulate a police officer inspecting parked cars. The class's responsibilities are as follows:
  - To know the police officer's name and badge number
  - To examine a `ParkedCar` object and a `ParkingMeter` object, and determine whether the car's time has expired
  - To issue a parking ticket (generate a `ParkingTicket` object) if the car's time has expired

Write a program that demonstrates how these classes collaborate.

9. Geometry Calculator

Design a `Geometry` class with the following methods:

- A static method that accepts the radius of a circle and returns the area of the circle. Use the following formula:
  $Area = \pi r^2$
  Use `Math.PI` for π and the radius of the circle for r.
- A static method that accepts the length and width of a rectangle and returns the area of the rectangle. Use the following formula:
  $Area = Length \times Width$
- A static method that accepts the length of a triangle's base and the triangle's height. The method should return the area of the triangle. Use the following formula:
  $Area = Base \times Height \times 0.5$

The methods should display an error message if negative values are used for the circle's radius, the rectangle's length or width, or the triangle's base or height.

Next, write a program to test the class, which displays the following menu and responds to the user's selection:

```
Geometry Calculator
1.  Calculate the Area of a Circle
2.  Calculate the Area of a Rectangle
3.  Calculate the Area of a Triangle
4.  Quit

Enter your choice (1-4):
```

Display an error message if the user enters a number outside the range of 1 through 4 when selecting an item from the menu.

Write a statement that may appear in a subclass that calls this method, passing 10 as an argument.

6. A superclass has the following abstract method:

```
public abstract int getValue();
```

Write an example of a `getValue` method that can appear in a subclass.

7. Write the first line of the definition for a `Stereo` class. The class should extend the `SoundSystem` class, and it should implement the `CDplayable`, `TunerPlayable`, and `CassettePlayable` interfaces.

8. Write an interface named `Nameable` that specifies the following methods:

```
public void setName(String n)
public String getName()
```

Short Answer

1. What is an "is-a" relationship?
2. A program uses two classes: `Animal` and `Dog`. Which class is the superclass and which is the subclass?
3. What is the superclass and what is the subclass in the following line?

```
public class Pet extends Dog
```

4. What is the difference between a protected class member and a private class member?
5. Can a subclass ever directly access the private members of its superclass?
6. Which constructor is called first, that of the subclass or the superclass?
7. What is the difference between overriding a superclass method and overloading a superclass method?
8. Reference variables can be polymorphic. What does this mean?
9. When does dynamic binding take place?
10. What is an abstract method?
11. What is an abstract class?
12. What are the differences between an abstract class and an interface?

Programming Challenges

1. `Employee` and `ProductionWorker` Classes

Design a class named `Employee`. The class should keep the following information in fields:

- Employee name
- Employee number in the format XXX–L, where each X is a digit within the range 0–9 and the L is a letter within the range A–M.
- Hire date

Write one or more constructors and the appropriate accessor and mutator methods for the class.

Next, write a class named `ProductionWorker` that extends the `Employee` class. The `ProductionWorker` class should have fields to hold the following information:

- Shift (an integer)
- Hourly pay rate (a `double`)

The workday is divided into two shifts: day and night. The shift field will be an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the classes by writing a program that uses a `ProductionWorker` object.

2. `ShiftSupervisor` Class

In a particular factory, a shift supervisor is a salaried employee who supervises a shift. In addition to a salary, the shift supervisor earns a yearly bonus when his or her shift meets production goals. Design a `ShiftSupervisor` class that extends the `Employee` class you created in Programming Challenge 1. The `ShiftSupervisor` class should have a field that holds the annual salary and a field that holds the annual production bonus that a shift supervisor has earned. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a `ShiftSupervisor` object.

3. `TeamLeader` Class

In a particular factory, a team leader is an hourly paid production worker that leads a small team. In addition to hourly pay, team leaders earn a fixed monthly bonus. Team leaders are required to attend a minimum number of hours of training per year. Design a `TeamLeader` class that extends the `ProductionWorker` class you designed in Programming Challenge 1. The `TeamLeader` class should have fields for the monthly bonus amount, the required number of training hours, and the number of training hours that the team leader has attended. Write one or more constructors and the appropriate accessor and mutator methods for the class. Demonstrate the class by writing a program that uses a `TeamLeader` object.

4. `Essay` Class

Design an `Essay` class that extends the `GradedActivity` class presented in this chapter. The `Essay` class should determine the grade a student receives for an essay. The student's essay score can be up to 100 and is determined in the following manner:

Grammar: 30 points
Spelling: 20 points
Correct length: 20 points
Content: 30 points

Demonstrate the class in a simple program.

Although there are complex encryption techniques, you should come up with a simple one of your own. For example, you could read the first file one character at a time, and add 10 to the character code of each character before it is written to the second file.

myCodeMate

8. File Decryption Filter

Write a program that decrypts the file produced by the program in Programming Challenge 7. The decryption program should read the contents of the coded file, restore the data to its original state, and write it to another file.

9. `TestScores` Modification for Serialization

Modify the `TestScores` class that you created for Programming Challenge 1 to be serializable. Write a program that creates an array of at least five `TestScore` objects and serializes them. Write another program that deserializes the objects from the file.

10. Exception Project

VideoNote
The Exception Project Problem

This assignment assumes you have completed Programming Challenge 1 of Chapter 11 (`Employee` and `ProductionWorker` Classes). Modify the `Employee` and `ProductionWorker` classes so they throw exceptions when the following errors occur:

- The `Employee` class should throw an exception named `InvalidEmployeeNumber` when it receives an employee number that is less than 0 or greater than 9999.
- The `ProductionWorker` class should throw an exception named `InvalidShift` when it receives an invalid shift.
- The `ProductionWorker` class should throw an exception named `InvalidPayRate` when it receives a negative number for the hourly pay rate.

Write a test program that demonstrates how each of these exception conditions works.

```
            return myMethod(num - 1) + num;
        }
    }
```

6. Convert the following iterative method to one that uses recursion:

```
public static void sign(int n)
{
    while (n > 0)
    {
        System.out.println("No Parking");
        n--;
    }
}
```

7. Write an iterative version (using a loop instead of recursion) of the `factorial` method shown in this chapter.

Short Answer

1. What is the difference between an iterative algorithm and a recursive algorithm?
2. What is a recursive algorithm's base case? What is the recursive case?
3. What is the base case of each of the recursive methods listed in Algorithm Workbench 3, 4, and 5?
4. What type of recursive method do you think would be more difficult to debug: one that uses direct recursion or one that uses indirect recursion? Why?
5. Which repetition approach is less efficient: a loop or a recursive method? Why?
6. When recursion is used to solve a problem, why must the recursive method call itself to solve a smaller version of the original problem?
7. How is a problem usually reduced with a recursive method?

Programming Challenges

1. Recursive Multiplication

Write a recursive function that accepts two arguments into the parameters `x` and `y`. The function should return the value of `x` times `y`. Remember, multiplication can be performed as repeated addition as follows:

7 * 4 = 4 + 4 + 4 + 4 + 4 + 4 + 4

2. `isMember` Method

Write a recursive `boolean` method named `isMember`. The method should accept two arguments: an array and a value. The method should return `true` if the value is found in the array, or `false` if the value is not found in the array. Demonstrate the method in a program.