APCS Lab 3 – Intro to classes

<u>80% Level</u>

1.

a) Add a method to the Car class, call it **changePrice(),** which will prompt the user for a new price of a particular instance of Car. Test your modified class.

b) Add the attribute *year*, which will be initialized as an integer argument of the constructor. Test your modified class.

2. Write a class called **BankAccount**.

It will have the following attributes:

balance	holds the amount currently in the account
annualRate	holds the annual interest rate in decimal form

It will have the following methods:

deposit(double a)	deposits amount into the account
withdraw(double a)	withdraws amount from the account
addInterest( )	calculates new balance, adding interest
getInterest( )	requests user for interest rate
getBalance( )	fetches the present balance for the account
printBalance()	prints the present balance

The constructor will pass one argument: the initial balance.

Test your class by writing a **BankAccountTest** program that sets up instances of accounts for three different people (e.g. Peter, Paul and Mary) and tests all the methods.

3. Write a class called **Employee**

It will have the following attributes:

```
    String name
    int age
    String job
    double salary
```

It will have the following methods:

```
/* Assign the age of the Employee  to the variable age.*/
   public void empAge(int empAge)

   /* Assign the designation to the attribute job.*/
   public void empJob(String empJob)

   /* Assign the salary to the attribute salary.*/
   public void empSalary(double empSalary)

   /* Print the Employee details: name, job, age and salary */
   public void printEmployee()

   /* Get the employee salary */
   public double getEmpSalary()

   /* Get the employee salary */
   public int getEmpAge()

   /* Get the employee job */
   public String getEmpJob()
```

The constructor will pass one argument: the name of the employee.
Test your class by writing a **EmployeeTest** program that sets up instances of Employee for two different people and tests all the methods.

4. Implement a class **SodaCan** with methods **getSurfaceArea( )** and **getVolume( ).** In the constructor, supply the height and radius of the can.

5. Implement a class **Student**. For the purpose of this exercise, a student has a name and a total quiz score. Supply an appropriate constructor, attributes and methods **getName( ), addQuiz(int score), getTotalScore( ),** and **getAverageScore( ).**

6. Implement a class **RoachPopulation** that simulates the growth of a roach population. The constructor argument will take the size of the initial roach population. The **waitPeriod( )** method simulates a period in which the roach population doubles. The **spray( )** method simulates spraying with an insecticide, which reduces the population by 10%. The **getRoaches( )** method returns the current number of roaches. Implement the class and a test program that simulates a kitchen that starts out with 10 roaches. The test program will wait, spray, display the roach count, repeating 3 times.

90% Level

7. Implement a class **Fraction** which will multiply two fractions. (Results need not be reduced – that's for a future lab.)
The constructor will pass two integers, the numerator and the denominator of a fraction.
Create the following methods:
public int getNum()
public int getDenom()
public Fraction multiplyFraction(Fraction a)
public void dispFraction()
and use the following attributes:
private int num;
private int denom;

The test program for the **Fraction** class is as follows:

```
/**
A class to test the Fraction class
*/

public class FractionTest
{
        public static void main(String[] args)
        {
                Fraction f1 = new Fraction(2,5);
                Fraction f2 = new Fraction(3,4);
                Fraction r = new Fraction(1,1);
                r = f1.multiplyFraction(f2);
                r.dispFraction();
        }
}
```

<u>100% Level</u>

8. Extend the Fraction class to include the methods
public Fraction divideFraction(Fraction a)
public Fraction addFraction(Fraction a)
public Fraction subtractFraction(Fraction a)

Also, extend the test program to ask for user input for the numerator and denominator of both the first and second fractions.