Com1101 Exam 1 – Winter 2003

Name: _____

Student Id (last 4 digits) : _____

• Write down the answers in the space provided. Use what you know from the lectures, labs, and the assigned readings.
• Please remember that the phrase "develop a function" means to design a function according to one of the recipes. You are *not* required to provide a template unless the problem specifically asks for one.
• You may obtain a maximum of 60 points.

Good luck.

Problem	Points
1	
2	
3	
4	
Total (out of 60)	
Base	60

15 Points

1 Problem

This problem consists of several, independent parts. In order to solve one part, you do not need to know the solution for any other part.

1. Develop the function `grossPay` that computes the pay for an employee, based on the hours worked and the hourly pay rate.

```
/* Compute the gross pay, given hours worked
 * and the hourly pay rate     */
double grossPay(double hours, double rate){
    return (hours * rate);
}

/* Test the function grossPay */
void TestGrossPay(){

    testHeader("grossPay(double hours, double rate)");

    expected(150.00);
    actual(grossPay(10.0, 15.0));

    expected(480.00);
    actual(grossPay(40.0, 12.0));
}
```

2. Develop the function computeTax which computes the tax on the weekly income according to the following table:

income	tax rate
under $200	10%
under $500	20%
$500 and over	30%

You must use a template here.

```
/* Compute the tax on given income, using the tax table */
double computeTax(double income){
    /* Template:
        ... income < 200 ...
        ... income < 500 ...
        ... else ...            */

    if (income < 200)
        return (income * 0.10);
    if (income < 500)
        return (income * 0.20);
    else
        return (income * 0.30);
}

/*****************************************************/
/* Test the function computeTax */
void TestComputeTax(){
    testHeader("computeTax(double income)");

    expected(18.00);
    actual(computeTax(180.00));

    expected(46.00);
    actual(computeTax(230.00));

    expected(180.00);
    actual(computeTax(600.00));
}
```

3. Given these functions, develop the function `paycheck` which computes the net pay (i.e. gross pay minus the tax).

```
/* Compute the net pay, given hours worked
 * and the hourly rate, using tax table                    */
double payCheck(double hours, double rate){
    return (grossPay(hours, rate) -
            computeTax(grossPay(hours, rate)));
}

/*********************************************************/
/* Test the function payCheck */
void TestPayCheck(){
    testHeader("payCheck(double hours, double rate)");

    expected(135.00);
    actual(payCheck(10.0, 15.0));

    expected(384.00);
    actual(payCheck(40.0, 12.0));

    expected(588.00);
    actual(payCheck(40.0, 21.0));
}
```

10 Points

2 Problem

Design the `class Tea`, which represents a tea can in the grocery store. The relevant information is the brand name, the weight of the can, given in ounces, and the price of the can, given in cents.

```
class Tea {
    /*-----------------------------------------------------
     The member data
    -----------------------------------------------------*/
    /* Brand name of the tea */
    String name;

    /* Weight in ounces      */
    double weight;

    /* Price of the tea      */
    int price;

    /*-----------------------------------------------------
     The constructor
    -----------------------------------------------------*/
    Tea(String aName, double aWeight, int aPrice){
        this.name   = aName;
        this.weight = aWeight;
        this.price  = aPrice;                    }

    /*****************************************************/
    /*-----------------------------------------------------
     Define a Tea object and print the member data values.
    -----------------------------------------------------*/
    void TestTea(){

        println("\nDefine Tea objects, print member data values");
        println(new Tea("Earl Grey", 10.0, 300));
        println(new Tea("Jasmine", 8.0, 320));
    }
```

1. Develop the method `unitPrice`, which computes the price per ounce of this grocery item.

```
/*-------------------------------------------------------
 Purpose:   to compute the unit price of this tea

 Template:  ... this.name...this.weight ... this.price ...
-------------------------------------------------------*/
double unitPrice(){
    return this.price/this.weight;
}

/*******************************************************/
/*-------------------------------------------------------
Test the method unitPrice.
-------------------------------------------------------*/
void TestUnitPrice(){

    Tea egt = new Tea("Earl Grey", 10.0, 300);
    Tea jas = new Tea("Jasmine", 8.0, 320);

    expected(30.0);
    actual  (egt.unitPrice());

    expected(40.0);
    actual  (jas.unitPrice());
}
```

2. Develop the method `isCheaperThan`, which determines whether the unit price is lower than some given price. Use a template.

```
 /*-------------------------------------------------------
  Purpose:   to determine whether the unit price of this tea
             is less than some given price

  Template:
  ...this.name...this.weight...this.price...this.unitPrice()
  ...aPrice
-------------------------------------------------------------
boolean isCheaperThan(int aPrice){
     return this.unitPrice() < aPrice;
 }

 /*********************************************************/
 /*-------------------------------------------------------
 Test the method isCheaperThan.
 -------------------------------------------------------*/
 void TestIsCheaperThan(){

     Tea egt = new Tea("Earl Grey", 10.0, 300);
     Tea jas = new Tea("Jasmine", 8.0, 320);

     expected(true);
     actual  (egt.isCheaperThan(35));

     expected(false);
     actual  (jas.isCheaperThan(35));
 }
```

3. Draw the UML diagram for this class.

15 Points

3 Problem

Given the UML diagram of classes `Person` and `Address` as shown:

1. Develop the method sameZip in the class Address that determines whether the address is in some zip code. Use template to develop this method.

   The `sameZip` method.

```
/* IN THE CLASS Address ------------------------------*/
/*-----------------------------------------------------
 Purpose:   to determine whether the zip code for this address
            is the same as the given zip code

 Template:  ... this.city...this.zip ... aZip ...
-----------------------------------------------------*/
boolean sameZip(int aZip){
    return (this.zip == aZip);
}

/* IN THE TEST SUITE ---------------------------------*/
/*****************************************************/
/*-----------------------------------------------------
 Test the method sameZip.
-----------------------------------------------------*/
void TestSameZip(){

    Address grey = new Address("Houston", 30057);
    Address jas  = new Address("Huntsville", 32054);

    expected(true);
    actual  (grey.sameZip(30057));

    expected(false);
    actual  (jas.sameZip(30057));
}
```

2. Develop the test suite that defines two objects in the class `Person` and performs two tests to determine whether either of these `Person`s lives in some zip code.

```
/*-------------------------------------------------------
 Test to determine whether a person lives in the given zip code.
-------------------------------------------------------*/
void TestZip(){

    Person earl =
        new Person("Earl Grey", new Address("Houston", 30057));

    Person jas  =
        new Person("Jasmine Joy", new Address("Miami", 42054));

    expected(true);
    actual  (earl.addr.sameZip(30057));

    expected(false);
    actual  (jas.addr.sameZip(30057));
}
```

20 Points

4 Problem

Given a UML class hierarchy to represent items in the garden store, develop the classes `Seedlings`, `Vegetables`, and `Flowers` as specified in the UML diagram.