Instructions: This homework is intended to test your skills on handling objects and classes in Java. You are free to discuss the problems with your classmates and instructor. However, it is not allowed to share code. Please indicate the source clearly if you take any help from internet and write sufficient comments to demonstrate your understanding. You will be benefited most if you try to solve these problems by yourself. Late submissions are generally not accepted.

Your Java source files should be placed in a package named “<firstname><lastname>hw3”. For example, if your first name and last name are “John” and “Smith” respectively, your package name should be johnsmithhw3. Meaning, your source files should be placed in a folder named "johnsmithhw3" and each of the source files should have the following package declaration at the beginning:

```
package johnsmithhw3;
```

Netbeans IDE can do this automatically for you. You just need to create a new package with the name as described above, and then create your classes in that package. Finally you need to locate the package folder in the project subdirectory.

You should submit a zip file containing the package folder that have all your source files (Pizza.java, PizzaOrder.java and Money.java).

Problem 1 (60 points):

Programming project 4.12 from textbook asks to write a class PizzaOrder that uses the class pizza from programming project 4.11. Programming project 5.8 asks to add more functionalities to the PizzaOrder class. Do programming projects 4.11, 4.12 and 5.8, and submit the final pizzaOrder and pizza classes with the main function suggested in programming project 5.8.

Programming Project 4.11 (page 256):

Create a class named Pizza that stores information about a single pizza. It should contain the following:

- Private instance variables to store the size of the pizza (either small, medium, or large), the number of cheese toppings, the number of pepperoni toppings, and the number of ham toppings.
- Constructor(s) that set all of the instance variables.
- Public methods to get and set the instance variables.
- A public method named calcCost() that returns a double that is the cost of the pizza.
  
  Pizza cost is determined by:
  Small: $10 + $2 per topping
  Medium: $12 + $2 per topping
  Large: $14 + $2 per topping

- A public method named getDescription() that returns a String containing the pizza size, quantity of each topping, and the pizza cost as calculated by calcCost().

Write test code to create several pizzas and output their descriptions. For example, a large pizza with one cheese, one pepperoni and two ham toppings should cost a total of $22.
Programming Project 4.12 (page 256):
This programming project extends Programming Project 4.11. Create a PizzaOrder class that allows up to three pizzas to be saved in an order. Each pizza saved should be a Pizza object as described in Programming Project 4.11. In addition to appropriate instance variables and constructors, add the following methods:

- public void setNumPizzas(int numPizzas) — sets the number of pizzas in the order. numPizzas must be between 1 and 3.
- public void setPizza1(Pizza pizza1) — sets the first pizza in the order.
- public void setPizza2(Pizza pizza2) — sets the second pizza in the order.
- public void setPizza3(Pizza pizza3) — sets the third pizza in the order.
- public double calcTotal() — returns the total cost of the order.

Write a main method to test the class. The setPizza2 and setPizza3 methods will be used only if there are two or three pizzas in the order, respectively. Sample code illustrating the methods is shown below. Note that first three lines are incomplete.

You must complete them as part of the Programming Project.

```java
Pizza pizza1 = new Pizza("Medium", "Cheese", 1, 0, 0); // Code to create a medium pizza, 1 cheese, 2 pepperoni
Pizza pizza2 = new Pizza("Large", "Cheese", 1, 1, 0); // Code to create a large pizza, 1 cheese, 1 ham
PizzaOrder order = new PizzaOrder(pizza1, pizza2); // Create an order
order.setNumPizzas(2); // 2 pizzas in the order
order.setPizza1(pizza1); // Set first pizza
order.setPizza2(pizza2); // Set second pizza
double total = order.calcTotal(); // Should be 18+20 = 38
```

Programming Project 5.8 (page 343):
Programming Project 4.12 asked you to create a PizzaOrder class that stores an order consisting of up to three pizzas. Extend this class with the following methods and constructor:

- public int getNumPizzas() — returns the number of pizzas in the order.
- public Pizza getPizza1() — returns the first pizza in the order or null if pizza1 is not set.
- public Pizza getPizza2() — returns the second pizza in the order or null if pizza2 is not set.
- public Pizza getPizza3() — returns the third pizza in the order or null if pizza3 is not set.
- A copy constructor that takes another PizzaOrder object and makes an independent copy of its pizzas. This might be useful if using an old order as a starting point for a new order.
- Write a main method to test the new methods. Changing the pizzas in the new order should not change the pizzas in the original order. For example,

```java
Pizza pizza1 = new Pizza("Medium", "Cheese", 1, 2, 0); // Code to create a medium pizza, 2 cheese, 2 pepperoni
PizzaOrder order1 = new PizzaOrder(pizza1); // Create an order
order1.setNumPizzas(2); // 2 pizzas in the order
order1.setPizza1(pizza1); // Set first pizza
order1.setPizza2(pizza2); // Set second pizza
double total = order1.calcTotal(); // Should be 18+20 = 38
PizzaOrder order2 = new PizzaOrder(order1); // Use copy constructor
order2.getPizza1().setNumCheeseToppings(3); // Change toppings
double total2 = order2.calcTotal(); // Should be 22 + 20 = 42
```

Note that the first three lines of code are incomplete. You must complete them as part of the Programming Project.

Problem 2 (40 points):

You need to know about the following things to start this problem.

- The methods equals and toString (page 205 in textbook)
- Accessor and Mutator methods (page 210 in textbook)
Now do the programming project 5.5 (page 341).

**Part One:** Define a class named `Money` whose objects represent amounts of U.S. money. The class should have two instance variables of type `int` for the dollars and cents in the amount of money. Include a constructor with two parameters of type `int` for the dollars and cents, one with one constructor of type `int` for an amount of dollars with zero cents, and a no-argument constructor. Include the methods `add` and `minus` for addition and subtraction of amounts of money. These methods should be static methods, should each have two parameters of type `Money`, and return a value of type `Money`. Include a reasonable set of accessor and mutator methods as well as the methods `equals` and `toString`. Write a test program for your class.

**Part Two:** Add a second version of the methods for addition and subtraction. These methods should have the same names as the static version but should use a calling object and a single argument. For example, this version of the `add` method (for addition) has a calling object and one argument. For example, this version of the `add` method (for addition) has a calling object and one argument. So `m1.add(m2)` returns the result of adding the `Money` objects `m1` and `m2`. Note that your class should have all these methods; for example, there should be two methods named `add`.

**Alternate Part Two:** Add a second version of the methods for addition and subtraction. These methods should have the same names as the static version but should use a calling object and a single argument. The methods should be `void` methods. The result should be given as the changed value of the calling object. For example, this version of the `add` method (for addition) has a calling object and one argument. Therefore,

```java
m1.add(m2);
```

changes the values of the instance variables of `m1` so they represent the result of adding `m2` to the original version of `m1`. Note that your class should have all these methods; for example, there should be two methods named `add`. 