

Declare and implement the classes described below.

```
#####  
#####
```

BASE Class Name: Item
Filenames: item.h/item.cpp

Private Members:
 string name_
 --holds the name of the item
 unsigned int value_
 --holds the value of the item

Public Members:
 Constructor
 --two parameters, one for each private member variable
 --defaults name_ to "item"
 --defaults value_ to 0

 Destructor
 --does nothing
 --must be virtual (add virtual keyword before the destructor)

Accessors and Mutators for the 2 private variables
 --use the naming scheme we have been using all semester

 string ToString()
 --returns a string containing name_ and value_
 Format -- name_, \$value_
 EXAMPLE -- shoes, \$25

```
#####  
#####
```

DERIVED Class Name: FoodItem (base Item)
Filenames: food_item.h/food_item.cpp

Private Members:
 unsigned int calories_
 --holds a calorie count
 string unit_of_measure_
 --holds the unit of measure (such as "ounces")
 double units_
 --holds a count of how many units we have

Public Members:
 Constructor
 --five parameters, one for each private member variable and two for the
base class
 --defaults name_ to "fooditem"
 --defaults value_ to 0
 --defaults calories_ to 0
 --defaults unit_of_measure_ to "nouns"

```

--defaults units_ to 0

Destructor
--does nothing
--must be virtual (add virtual keyword before the destructor)

Accessors and Mutators for the 3 private variables
--use the naming scheme we have been using all semester

string ToString()
--returns a string containing name_, value_, units_, unit_of_measure_,
and calories_
    (uses Item::ToString in its implementation)
    units_ formatted to exactly two decimal places
    Format -- name_, $value_, units_ unit_of_measure_, calories_
calories
    EXAMPLE -- cookie, $1, 2.50 cookie(s), 250 calories

#####

DERIVED Class Name: MagicItem (base Item)
Filenames: magic_item.h/magic_item.cpp

Private Members:
    string description_
    --holds a description of the magic item
    unsigned int mana_required_
    --holds the amount of mana required to use the magic item

Public Members:
    Constructor
    --four parameters, one for each private member variable and two for the
base class
    --defaults name_ to "magicitem"
    --defaults value_ to 0
    --defaults description_ to "no description"
    --defaults mana_required_ to 0

    Destructor
    --does nothing
    --must be virtual (add virtual keyword before the destructor)

    Accessors and Mutators for the 2 private variables
    --use the naming scheme we have been using all semester

    string ToString()
    --returns a string containing name_, value_, description_, and
mana_required_
        (uses Item::ToString in its implementation)
        Format -- name_, $value_, description_, requires mana_required_ mana
        EXAMPLE -- hat, $10, made of felt, requires 2 mana

```


#####