

OBJECT-ORIENTED PROGRAMMING**3rd Homework Assignment****Due on: April 17, 2006**

The Banking System Regulation authority has allowed to open account in other currencies such as USD and Euro. Make necessary changes in the classes so that people can have accounts in other currencies. ATM Machines can only serve in YTL. If the customer's account is in currency other than YTL, then ATM machine gives money in YTL and charge the amount, first by converting it to the currency the customer use, to the customer's account. Consider the following application codes in your design:

```

CentralBank *centralBank= new CentralBank() ; // should cause compile error
CentralBank *pCentralBank= CentralBank::getInstance() ; // this is OK
Currency::setCentralBank(pCentralBank); // Central Bank specifies the Exchange Rates
pCentralBank->setExchangeRate(EXC_RATE_USD,1.35); // 1USD=1.35YTL
pCentralBank->setExchangeRate(EXC_RATE_EURO,1.72); // 1Euro=1.72YTL
double euro_rate= pCentralBank->getExchangeRate(EXC_RATE_EURO); // should return 1.72
Currency *cur_in_usd = new USD(1250.3) ; // Create 1250.3 USD
Currency *cur_in_euro = new Euro(5703.75) ; // Create 5703.75 Euro
double amount= cur_in_euro->getAmount() ;
cur_in_euro->addAmount(-10.3) ; // now holds 5693.45 Euro
cur_in_usd->addAmount(7.13) ; // now holds 1257.43 USD
YTL *cur_in_ytl= cur_in_usd->convertToYTL() ; // cur_in_ytl holds 1.35 * 1257.43 amount in YTL
*cur_in_ytl=*cur_in_euro; // now cur_in_ytl holds 5693.45 * 1.72 YTL
delete cur_in_ytl ;
delete cur_in_euro;
delete cur_in_usd;

```

Preparation:

- Download the file <http://www.cs.itu.edu.tr/~kurt/Courses/blg332e/hw3.zip> to your local disk and uncompress it.
- First design and implement Currency and related classes and implement them.
- Then make necessary changes in existing classes with which you are familiar from the 2nd homework. Your solution also should be backward compatible with the application code, i.e. BankApp.cpp, that is provided in the 2nd the homework.

Testing:

- Write a unit test for Currency classes.
- Write an application code which tests all use cases.

How to submit:

- Archive hw3 directory into a zip file named as *your_student_id.zip*.
- **Use the following link to submit your homework:**
http://www.ce.itu.edu.tr/Members/kurt/blg332hw3/assignment_submit_form

IMPORTANT

- Academic dishonesty including but not limited to cheating, plagiarism, collaboration is unacceptable and subject to disciplinary actions. Any student found guilty will have grade F. Assignments are due in class on the due date. Late assignments will generally not be accepted. Any exception must be approved. Approved late assignments are subject to a grade penalty.