

**EGN 3211**  
**Engineering Analysis & Design**  
**Spring 2006**

**Home Work Schedule**

<b>HW#</b>	<b>Due</b>	<b>Points</b>	<b>Exercises</b>
1	1/23/06	1%	Hello world program
2	2/06/06	1%	Compound interest program.
3	3/27/06	3%	Part 1: Functions, IO, arrays, loops
4	4/10/06	3%	Part 2: Files
5	4/24/06	2%	Part 3: Structures
EC	4/24/06	3%	Part 4: Objects and classes (extra credit)

For the Tuesday-Thursday class the HW is due the next day.

All homework must include:

- A Cover page with the following information
  - Course number (EGN 3210)
  - Your name
  - Home work number
- Short description of the work done.
- Source code with comments.
- Program output.

**Notes:**

- Comments on the source code are to explain what the code does. This will help the grader (TA) as well as your self understand how your code works and what each function does. Each function must have comments indicating exactly what that function does.
- Homework assignments are due at the BEGINING of the class meeting for the day it is due. Late HW receives a 1 time penalty of 10%. No HW will be accepted after the next HW is due (approximately 2 to 3 weeks). HW that does not compile or compiles but does not run or runs but there is no output to prove it will receives a grade of 50%. Therefore if the program does not run and its due it is better to keep it and get it to run before turning it in provided it does not exceed the next due date.
- Homework assignments are to be turned in to me (course instructor, Dr. Gonzalez) or put into my mailbox or under my office door (enr 211).
- You are required to keep a copy of all material incase it gets lost. You must also keep all graded material returned to you to prove it has been turned in on time.

## **Homework Assignment #1**

EGN 3210 Engineering Analysis & Design

Topic: Familiarity with the compiler and entering and running the program.

### **The Hello World! Program**

Enter the Hello World program and run it. Do not forget to print out the program listing and the output.

## **Homework Assignment #2**

EGN 3210 Engineering Analysis & Design

Topic: Familiarity with functions, assignments, loops, if and user input.

### **Compound Interest**

Write a program that computes the balance of a saving account at a bank after adding the compound interest. For example if I have \$100 in my account and I get 12% interest (per year) compounded monthly then at the end of 3 months I will have:

$$(((\$100 * 1.01) * 1.01) * 1.01) = \$103.03$$

Note that the monthly interest is  $\frac{12\%}{12} = 1\%$  and if I do not consider the fact that it is

compounded monthly I will simply have  $(100 * 1.03) = \$103 \neq 103.03$ .

Your program should do the following steps:

1. Your program is to ask the user for :
  - the initial savings amount
  - the interest rate
  - the number of months to compound.
2. If the user enters an invalid number you must tell the user to reenter the numbers and repeat the user input routine (ask the user again).
3. Then compute the balance at the end of the period the user specified. You may use a loop construct like the while or for loop instructions.
4. Print the results in an elegant way.

### Homework Assignment #3

EGN 3210 Engineering Analysis & Design

Topic: Familiarity with functions, loops, formatted output, user interface and arrays.

#### **Loan support part 1**

The following 3 assignments will result in a single program to be used by the loan department of a bank. It will create new loans, update them as payments are received, generate payment schedules, print loan information reports and close existing accounts.

To open an account each customer will need the following data: name, address, phone, SSN, age, initial loan amount, loan length, and credit score. You will then generate the following data: monthly payment, current balance and the interest rate. All this data, the one entered by the user and the data generated, must be stored in a data base implemented with an array. The data base of customers will be implemented as an array with each component of the array corresponding to a unique customer. You will need several arrays since there are several data types needed to store the customer's data.

For this assignment you will write the following 5 functions:

Get new account input

Find an empty array component.

Get name, address, phone, ssn, age from the user.

Get initial loan amount, loan length (n), credit score (CS) from the user.

Store this information in the data base (the arrays).

Open a new loan:

Given the initial loan amount, n, and CS compute the following:

Account number = next account number

Update next account number for next time.

$$\text{Interest rate} = i = 5\% + \frac{800 - CS}{100}$$

$$\text{Monthly payment} = \text{initial amount} * \left[ \frac{i(1+i)^n}{(1+i)^n - 1} \right]$$

Balance = initial amount;

Make a payment towards an existing account:

Get the actual payment amount ( *pay* ) and the account number

Find the account and change its bal as

$$\text{Balance} = (i + 1)\text{Balance} - \text{pay}$$

Close an existing account:

Get the account number from the user

If balance > 0 then

display "can not close account. Must pay balance"

else

blank out all strings, zero out all values, and mark that array component as available.

Generate a report of all of the loans.

Given the data base print each loan data on a single line.

Print all the loans (1 loan per line)

To test this program run it, open several accounts, make several payments, and display all data using the report function to compare with the theoretical values. Try closing an account by making all the payment.

Example:

Say the user enters:

Name: Bill Smith

Address: 4000 Central Florida Blvd. Orlando FL. 32816

Phone: (407) 823-2000

SSN: 123-45-6789

Age: 25

Loan amount  $p = 100000$

Loan length  $n = 30$  (in years)

Credit score  $CS = 700$

Then compute:

$$i = 5\% + \frac{800 - CS}{100} = 5\% + \frac{800 - 700}{100} = 5\% + 1\% = 6\%$$

but since the loan is compounded monthly this yields a monthly interest of

$$i = \frac{6\%}{12} = \frac{0.06}{12} = .005 \text{ Also the loan length is in years so it must also be multiplied by 12}$$

to get it in terms of number of months. So  $n = 30(12) = 360$

Then compute:

$$A = P \left[ \frac{i(1+i)^n}{(1+i)^n - 1} \right] = 100000 \frac{.005(1+0.005)^{360}}{(1+0.005)^{360} - 1} = 100000(.005995) = \$599.55$$

And

$$Bal = P = \$100,000$$

Finally compute:

$$Acct = Acct++;$$

Say the user makes a payment of \$650.00 then

$$Bal = (i + 1)Bal - pay = (1.005)100000 - 650 = \$99,850.00$$

## **Homework Assignment #4**

EGN 3210 Engineering Analysis & Design

Topic: Familiarity with files and arrays.

### **Functions and I/O**

In this homework you will add functions to create a payment schedule and save the data base to a file, and read in the data base from the file. To run the program you will create a menu and ask the user to select a task. Then run the corresponding task. The payment schedule is a report that shows how much is in the account after each payment. This is basically the balance after deducting the interest portion of each payment. The file is to save the data base so that the next time one runs the program the data base can be reloaded. The file will have each line corresponding to a loan, just like the report generated in the previous homework. The only difference is that the report is saved to a file instead of to the monitor and the headings need not be saved.

### **Save the data base to a file**

The data associated with a file is the file pointer. The following are some code that can be used to write and read from the file:

```
FILE *fp;
```

To open file for writing: the file is created and is named bill.dat. You may name the file to anything you like. The w means the file is created and will be used for writing.

```
fp = fopen("bill.dat","w");
```

To write to the file is the same as to print or write to the monitor except the output is directed to the file. You will use the file version of printf, fprintf, and pass to it as the first argument the file pointer, fp. The following line will print hello and a number to the file.

```
fprintf(fp,"Hello %d \n",x);
```

To read from the file you must open the file for reading. You need to use the file name you used to save the file so that the operating system can locate the data base. If you want to save the program after opening it for reading you need to close the file and reopen it for writing.

```
fp = fopen ("bill.dat","r");
```

Then to read from the file you can use the file version of scanf, fscanf, and pass it the file pointer, fp. The following reads in an integer from the file and saves in into the variable x.

```
fscanf(fp,"%d",&x);
```

At the end of the program you need to close the file. Use the following statement to do that.

```
fclose(fp);
```

You can use the same loop you used to create a report but redirect the output to a file. Do not save the heading information since this is not part of the data.

### Create a menu

The menu is a simple function that asks the user to select something. An example is below.

```
int    HowToTravelMenu( )
    {
    int    x;

    printf("    1.    Ride a bike    :\n");
    printf("    2.    Drive a car    :\n");
    printf("    3.    Ride the train    :\n");
    printf("    9.    Quit    :\n");
    printf("    \nPlease make a choice    : ");
    scanf("%d", &x);

    return x;
    }
```

### Create payment schedule

Create a report with the following fields: beginning balance, payment, interest paid, principle paid, and ending balance.

The beginning balance in the first line is the current balance (bal).

```
While beginning balance > 0
    if (i+1)P > A then
        Payment = A
    else
        Payment = (i+1)P
    intereste payed = iP
    principlly payed = A-iP
    ending balance = (i+1)P - A
    beginning balance = ending balance
    print all these number on a single line.
```

Example:

Consider the previous example where:

$$i = 0.005,$$

$$A = \$599.00$$

$$\text{Bal} = \$100,000$$

Now to create a payment schedule we will do the following:

Compute:

$$\text{Beginning balance is } \$99,850.00$$

$$\text{Interest paid is } iP = 0.005(99,850.00) = \$499.25$$

$$\text{Principle paid is } A - iP = 599.55 - 499.25 = \$100.30$$

$$\text{Ending balance is } (1+i)P - A = (1.005)99850.00 - 599.55 = \$99,749.70$$

So the first line will be

\$99,850	\$499.25	\$100.30	\$99,749.70
----------	----------	----------	-------------

The second line will be:

$$\text{Beginning balance is } \$99,749.70$$

$$\text{Interest paid is } iP = 0.005(99749.70) = \$498.75$$

$$\text{Principle paid is } A - iP = 599.55 - 498.75 = \$100.80$$

$$\text{Ending balance is } (1+i)P - A = (1.005)99749.70 - 599.55 = \$99,648.90$$

And the second line will be:

\$99,749.70	\$498.75	\$100.80	\$99,648.90
-------------	----------	----------	-------------

And so on until  $(i+1)P > A$  then the payment of  $A = 599.55$  becomes  $(i+1)P$  instead of  $A$  and the ending balance should become \$0.

The first part of the report should look like:

Beginning Bal.	Interest paid	Principle paid	Ending bal.
\$99,850.00	\$499.25	\$100.30	\$99,749.70
\$99,749.70	\$498.75	\$100.80	\$99,648.90

and so on.

For this homework you will write the 4 functions, read the data base from the file, save the database to the file, generate the payment schedule, and the menu function. The program will continuously run in a loop. Use the following algorithm:

```
While (!done)
{
```

```
selection = call the menu function
switch (selection)
{
    case 1:
        perform task associated with 1.
        break;
    case 2:
        perform task associated with 2.
        break;
    .
    .
    .
    case 9:
        done = TRUE;
        break;
    default:
        Printf("Invalid selection");
}
```

To test the functions start the program, add several loans and save the data. Next start the program again and load the data base and generate a payment schedule.

## **Homework Assignment #5**

EGN 3210 Engineering Analysis & Design

Topic: Structures.

### **Structures**

In this homework you will change the data structures from a set of arrays to an array of structures. You must change the way your code access the data since the data structures changed. The program is to generate output and operate the same as in the previous homework.

## **Homework Assignment #6 (Extra Credit)**

EGN 3210 Engineering Analysis & Design

Topic: classes and objects.

### **Classes and Objects**

In this homework you will change the data structures from an array of structures to an array of objects.

A: You are to change the record structure to a class that has as private elements all of the data in a loan (name, address, phone, SSN, age, initial loan amount, loan length, credit score, monthly payment, current balance, and the interest rate). Add to this class, as public member functions, the functions to open a new loan, close an existing loan, make a payment, print the data in the loan, print a payment schedule, and a new function to initialize all of the private data with data passed to the function. This initialize function is to allow the object to get initialized with the data in the file. Add a constructor to initialize every element to “empty” or 0.

B: Change the array of structures to an array of record objects. Call each object’s initialize function to set its values using the data in the file.

C: To execute a function you need to get the account number then call the function from the corresponding object.