Most of the software engineers I know here in Bangalore have either already bought a house or are planning to buy one. The biggest incentives are perhaps the easy availability of home loans at interest rates far lower than that available to the previous generation and the tax-breaks one gets here in India on the principal and the interest paid on home loans. Our generation is also not averse to taking on a debt unlike the previous generation. In addition, many of us feel that it is better to take the plunge now and enjoy the comforts of your own house than to diligently save all the required money for years and then buy a house, only to find out that the dream was realised a bit too late in your life.  
  
When one takes a loan, a natural question that comes to mind is how much would the EMI (Equated Monthly Installment) be that one has to pay back to the bank every month. Finding out the EMI for different tenures of the home loan allows one to select the best tenure based on one's current and projected income and expenses and possibly other factors.  
  
What I find is that most software engineers would rather construct or borrow elaborate spreadsheets for this purpose or use a web-based EMI calculator rather than calculating it themselves. Some brave souls ask around for the formula to calculate EMIs and then try to write a programme that calculates the EMIs for them. I find it particularly appalling that very few bother to figure it out for themselves, especially since it involves elementary algebra that most of us have surely learnt in high school. It either reflects the creeping sloth and sloppiness in our generation or the rote learning and regurgitating of formulae that our system of education seems to encourage. In any case, I attempt to show in this post how simple it really is to figure out how to calculate EMIs on your own and to hopefully encourage my fellow engineers into applying the basic mathematics they learn in school to real-life problems themselves instead of looking around to see if someone else has solved it for them.  
  
For the sake of simplicity, assume that the loan is offered on a "monthly rest" basis. That is, the bank calculates the interest at the end of every month on the amount you still owe to the bank at the beginning of the month, adds it to the amount you already owe and then deducts your EMI from this to calculate the total amount you still owe to the bank at the beginning of the next month. Some banks offer loans on a "daily rest" basis, where the outstanding amount and the interest is recalculated every day, but you still pay back on a monthly basis. The older "annual rest" basis is no longer in use as far as I can tell. Note that it is easy to adapt the formula given here to the "daily rest" basis and that is, of course, left as an exercise for the reader.  
  
Suppose you take on a loan for **P** Rupees, the tenure of the loan is **n** months (for example, n=240 for a 20-year loan), the *monthly rate* of interest is **r** (usually calculated by dividing the annual rate of interest quoted by the bank by 12, the number of months in a year, and dividing that by 100 as the rate is usually quoted as a percentage) and **E** Rupees is the EMI you have to pay every month. Let us use Pi to denote the amount you still owe to the bank at the end of the i-th month. At the very beginning of the tenure, i=0 and P0=P, the principal amount you took on as a loan.  
  
At the end of the first month, you owe the bank the original amount P, the interest accrued at the end of the month r×P and you pay back E. In other words:

P1 = P + r×P - E

or to rewrite it slightly differently:

P1 = P×(1 + r) - E

Similarly, at the end of the second month the amount you still owe to the bank is:

P2 = P1×(1 + r) - E

or substituting the value of P1 we calculated earlier:

P2 = (P×(1 + r) - E)×(1 + r) - E

and once again expanding it and rewriting it slightly differently:

P2 = P×(1 + r)2 - E×((1 + r) + 1)

where "xy" denotes "x raised to the power y" or "x multiplied by itself y times". To make this look slightly simpler, we substitute "(1 + r)" by "t" and now it looks like this:

P2 = P×t2 - E×(1 + t)

Continuing in this fashion and calculating P3, P4, etc. we quickly see that Pi is given by:

Pi = P×ti - E×(1 + t + t2 + ... + ti-1)

At the end of n months (that is, at the end of the tenure of the loan), the total amount you owe to the bank should have become zero. In other words, Pn=0. This implies that:

Pn = P×tn - E×(1 + t + t2 + ... + tn-1) = 0

which means that:

P×tn = E×(1 + t + t2 + ... + tn-1)

We can simplify this further by noticing that we have a [geometric series](http://en.wikipedia.org/wiki/Geometric_progression) of n terms here with a common ratio of t and a scale factor of 1. The sum of such a series is given by "(tn - 1)/(t - 1)", which we substitute in the above equation to yield:

P×tn = E×(tn - 1)/(t - 1)

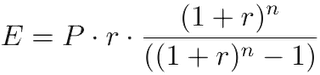
which can be rewritten as:

E = P×tn×(t - 1)/(tn - 1)

which can again be rewritten by substituting the value of t back as "(1 + r)" as:

E = P×r×(1 + r)n/((1 + r)n - 1)

and *this* is the formula for calculating your EMI. This formula can also be rendered more clearly as:



Suppose you take a loan from a bank of 10,00,000 Rupees for 15 years at 8.5% annual rate of interest calculated on a monthly rest basis. In that case, **P** = 10,00,000, **n** = 15×12 = 180 and **r** = (8.5/12)/100 = 0.0070833333. Putting these values into the formula given above gives us **E** = 9847.40 (approximately).  
  
When you write a programme to calculate your EMI using the formula given above, be careful to structure your computations to accommodate loss of precision and rounding errors. For more information, read "[What Every Computer Scientist Should Know About Floating-Point Arithmetic](http://cch.loria.fr/documentation/IEEE754/ACM/goldberg.pdf)" (PDF) by David Goldberg or section 4.2, "Floating Point Arithmetic", in "The Art of Computer Programming, Volume 2: Seminumerical Algorithms" (3rd Edition) by Donald Knuth.  
  
**Update (2006-12-19):** Clarified how "r" is calculated from the quoted interest rate and added an example to illustrate the application of the formula. Added references to Goldberg's article and Knuth's book for the caveats to bear in mind while using floating-point arithmetic.  
  
**Update (2007-02-01):** The only intent of this post was to show how basic mathematics can be used to calculate EMIs. I am just an ordinary computer programmer and not a financial consultant. Please stop asking me for advice on your loan plans!

1. **Five minutes to understanding reducing balance**

[Share](http://www.addthis.com/bookmark.php?v=250&username=pranjal26) |

Font[Larger](javascript:increaseFontSize();) | [Smaller](javascript:decreaseFontSize();)

 [Print](http://www.indianexpress.com/story_print.php?storyId=46983)

**MONIKA HALAN** **Posted: May 16, 2004 at 0000 hrs IST**

* Discussion
* Blogs

[Dhoni the midas - By Raghuvanshi](http://www.indianexpress.com/news/dhoni-the-midas/802872/#postComment)[Hi... - By mohan](http://www.indianexpress.com/news/hi.../802863/#postComment)[Hope bhaskaran like ... - By Sree Rama Sundaram](http://www.indianexpress.com/news/hope-bhaskaran-like-are-not-there-in-sc/802863/#postComment)[No taxes - By SRINIVAS](http://www.indianexpress.com/news/no-taxes/802863/#postComment)[Apex court's hearing... - By Ryk](http://www.indianexpress.com/news/apex-courts-hearing-on-ms.-knaimohnzi/802863/#postComment)[Stop paying taxes - By SRINIVASAN](http://www.indianexpress.com/news/stop-paying-taxes/802863/#postComment)[:) just for public - By Tarun](http://www.indianexpress.com/news/just-for-public/802863/#postComment)

http://static.indianexpress.com/frontend/iep/images/discussion_bot_bg1.gif

What is a ‘reducing balance’ loan?Suppose you took a Rs 1 lakh loan today at a rate of interest of 10 per cent for five years. You are to pay back Rs 20,000 of the principal and Rs 10,000 (10 per cent of the loan) every year. So you pay back Rs 30,000 every year. Over five years you pay back Rs 1.5 lakh. But notice, that the loan kept reducing over the five years as you paid back Rs 20,000 each year, yet you went on paying interest for five years, as if you had kept the Rs 1 lakh for the entire term.What if you paid an interest only on the amount you owed each year and not the entire one lakh? The first year you would pay Rs 10,000 as interest, the next year you would pay Rs 8,000 on a reduced principal of Rs 80,000 and so on, till the last year, you pay only Rs 2,000 as interest. Now you would have paid back Rs 1.3 lakh instead of Rs 1.5 lakh as in the earlier case. The first case is a situation of a loan that charges interest at a flat rate and the second case is when the interest is calculated on a ‘reducing balance’ or only on the amount of loan left to pay and not the entire loan amount. A flat 10 per cent is equal to a reducing balance at 6 per cent per annum.How often can the balance reduce?You can get a range of options in reducing balance loans. You get annual, quarterly, monthly, weekly and now daily rests. A ‘rest’ is jargon to indicate when the bank will recalculate the EMI based on the amount of loan paid back. Suppose you have a loan with an annual ‘rest’ then, though you pay a monthly instalment, your benefit kicks in only at year end. Meaning the bank gets free interest for 11 months. A monthly ‘rest’ will recognise the reduction in the loan amount on a monthly basis and a daily ‘rest’ will do it each day.What use is a daily ‘rest’ if the EMI is monthly?True, the loan instalments are usually on a monthly basis but new innovative products in the market make daily rests a possibility. For example, some home loan companies give an option to park extra funds in an account linked to the home loan account. The interest on the home loan is calculated keeping in mind the number of days the extra cash stays in that account and reducing the principal for those many days by the amount in the bank. A non-reducing balance loan is out of question today, settle for a monthly rest if you pay back each month or if there are options of continuous pre-payment of the loan, then ask for a daily rest.