Compound & Simple Interest Calculator

Mini- Project

Submitted in fulfilment of the requirement of University of Mumbai
For the Degree of

Engineering

in

(Computer Engineering)

by

1) Amey Thakur ID No: TU3F1819127 Roll No: B-50

2) Hasan Rizvi ID No: TU3F1819130 Roll No: B-51

Under the Guidance of

Pramila Mate Ma'am

Department of Computer Engineering
TERNA ENGINEERING COLLEGE
Plot no.12, Sector-22, Opp. Nerul Railway station,
Phase-11, Nerul (w), Navi Mumbai 400706
UNIVERSITY OF MUMBAI

Terna Engineering College

NERUL, NAVI MUMBAI

CERTIFICATE

This is to certify that

1) **Amey Thakur**
2) **Hasan Rizvi**

Has satisfactorily completed the requirements of the ***Operating System Laboratory Mini Project***

entitled

“Compound & Simple Interest Calculator”

As prescribed by the ***University of Mumbai*** *Under the guidance of*

Pramila Mate Ma’am

Guide **HOD** **Principle**

Index

Chapter **Page No.**

Chapter 1

ABSTRACT

The calculation of the principal accumulation value is the foundation of all financial calculations and plays a critical role in all financial activity such as investment, and financing. The theorem for calculations of the principal accumulation value of simple and compound interest is proposed. When a term is less than a period, the accumulated value of simple interest is more than that of compound interest; When a term is equal to a period, the accumulated value of simple interest is the same as that of compound interest; When a term is more than a period, the accumulated value of simple interest is less than that of compound interest, and the growth rate of the accumulated value of simple interest is less than that of compound interest. The theorem enriches the theory of interest. Finally, I strictly prove the theorem with Rolle's Theorem.

KEYWORDS

Principal accumulation value; Compound interest;

Simple interest;

Rolle's theorem;

Growth rate.

Chapter 2

INTRODUCTION

Interest is the cost of borrowing money, where the borrower pays a fee to the lender for using the latter's money. The interest, typically expressed as a percentage, can be either simple or compounded. Simple interest is based on the principal amount of a loan or deposit, while compound interest is based on the principal amount and the interest that accumulates on it in every period. Since simple interest is calculated only on the principal amount of a loan or deposit, it's easier to determine than compound interest.

Commands

- echo command

echo command in linux is used to display lines of text/string that are passed as an argument . This is a built in command that is mostly used in shell scripts and batch files to output status text to the screen or a file.

Syntax :

echo [option] [string]

Options of echo command

NOTE :- -e here enables the interpretation of backslash escapes

1. Input a line of text and display on standard output

$ echo Terna Engineering College

Outputs the following text:

Terna Engineering College

2. Declare a variable and echo its value. For example, Declare a variable of x and assign its value=10.

$ x=10

echo its value:

$ echo The value of variable x = $x

The value of variable x = 10

3. Using option '\b' – backspace with backslash interpreter '-e' which removes all the spaces in between.

$ echo -e "Amey \bThakur \bSE-Comps-B \b50 \bTerna \bEngineering \bCollege"

AmeyThakurSE-Comps-B50TernaEngineeringCollege

4. Using option '\n' – New line with backspace interpreter '-e' treats new line from where it is used.

$ echo -e "Amey \nThakur \nSE-Comps-B \n50 \nTerna \nEngineering \nCollege"

Amey

Thakur

SE-Comps-B

50

Terna

Engineering

College

5. Using option ‘\t‘ – horizontal tab with backspace interpreter ‘-e‘ to have horizontal tab spaces.

$ echo -e "Amey \tThakur \tSE-Comps-B \t50 \tTerna \tEngineering \tCollege"

Amey Thakur SE-Comps-B 50 Terna Engineering College

6. How about using the option new Line ‘\n‘ and horizontal tab ‘\t‘ simultaneously.

$ echo -e "\n\tAmey \n\tThakur \n\tSE-Comps-B \n\t50 \n\tTerna \n\tEngineering \n\tCollege"

Amey

Thakur

SE-Comps-B

50

Terna

Engineering

College

7. Using option ‘\v‘ – vertical tab with backspace interpreter ‘-e‘ to have vertical tab spaces.

$ echo -e "\vAmey \vThakur \vSE-Comps-B \v50 \vTerna \vEngineering \vCollege"

Amey

Thakur

SE-Comps-B

50

Terna

Engineering

College

8. How about using the option new Line ‘\n‘ and vertical tab ‘\v‘ simultaneously.

$ echo -e "\n\vAmey \n\vThakur \n\vSE-Comps-B \n\v50 \n\vTerna \n\vEngineering \n\vCollege"

Amey

Thakur

SE-Comps-B

50

Terna

Engineering

College

Note: We can double the vertical tab, horizontal tab and new line spacing using the option two times or as many times as required.

9. Using option ‘\r‘ – carriage return with backspace interpreter ‘-e‘ to have specified carriage return in output.

$ echo -e "Amey \rThakur SE-Comps-B 50 Terna Engineering College"

Thakur SE-Comps-B 50 Terna Engineering College

10. Using option ‘\c‘ – suppress trailing newline with backspace interpreter ‘-e‘ to continue without emitting a new line.

$ echo -e "Amey Thakur SE-Comps-B 50 \cTerna Engineering College"

Amey Thakur SE-Comps-B 50amy@amey:~$

11. Omit echoing trailing new line using option ‘-n‘.

$ echo -n "Amey Thakur SE-Comps-B 50 Terna Engineering College"

Amey Thakur SE-Comps-B 50 Terna Engineering Collegesamy@Amey:~/Desktop$

12. Using option ‘\a‘ – alert return with backspace interpreter ‘-e‘ to have sound alert.

$ echo -e "Amey Thakur SE-Comps-B 50 Terna \aEngineering College"

Note: Make sure to check Volume key, before firing.

13. Print all the files/folder using echo command (ls command alternative).

$ echo *

103.odt 103.pdf 104.odt 104.pdf 105.odt 105.pdf 106.odt 106.pdf 107.odt 107.pdf 108a.odt 108.odt 108.pdf 109.odt 109.pdf 110b.odt 110.odt 110.pdf 111.odt 111.pdf 112.odt 112.pdf 113.odt linux-headers-3.16.0-customkernel_1_amd64.deb linux-image-3.16.0-customkernel_1_amd64.deb network.jpeg

14. Print files of a specific kind. For example, let's assume you want to print all '.jpeg' files, use the following command.

$ echo *.jpeg

network.jpeg

15. The echo can be used with a redirect operator to output to a file and not standard output.

$ echo "Test Page" > testpage

Check Content

avi@tecmint:~$ cat testpage

Test Page

echo Options

Options	Description
-n	do not print the trailing newline.
-e	enable interpretation of backslash escapes.
\b	backspace
\\	backslash
\n	new line
\r	carriage return
\t	horizontal tab
\v	vertical tab

- Bash Read Bulletin

On Unix-like operating systems, read is a builtin command of the Bash shell. It reads a line of text from standard input and splits it into words. These words can then be used as the input for other commands.

Syntax:

read

read command without any option: The read command asks for the user's input and exit once the user provides some input.

- expr command

The expr command in Unix evaluates a given expression and displays its corresponding output. It is used for:

- Basic operations like addition, subtraction, multiplication, division, and modulus on integers.
- Evaluating regular expressions, string operations like substring, length of strings etc.

Syntax:

$expr expression

- esac statement

esac statement is to give an expression to evaluate and to execute several different statements based on the value of the expression. The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a default condition will be used.

Chapter 3

Implementation

- Simple Interest
  Simple interest is calculated using the following formula:

Simple Interest = $P \times r \times n$

where:

P = Principal amount

r = Annual interest rate

n = Term of loan, in years

Generally, simple interest paid or received over a certain period is a fixed percentage of the principal amount that was borrowed or lent. For example, say a student obtains a simple-interest loan to pay one year of their college tuition, which costs $18,000, and the annual interest rate on their loan is 6%. They repay their loan over three years. The amount of simple interest they pay is:

$3,240=$18,000×0.06×3

and the total amount paid is:

$21,240=$18,000+$3,240

For Example, if you borrow $1,000 at 7% simple interest for five years, you'll owe $350 in interest.

- Compound interest

Compound interest accrues and is added to the accumulated interest of previous periods; it includes interest on interest, in other words.

The formula for compound interest is:

Compound Interest = $P \times (1+r)^t - P$

where:

P=Principal amount

r=Annual interest rate

t=Number of years interest is applied

It is calculated by multiplying the principal amount by one plus the annual interest rate raised to the number of compound periods, and then minus the reduction in the principal for that year.

In the real world, simple interest is rarely used. When you deposit money into an interest-bearing account or take out a line of credit, the interest that accumulates is added to the principal, and the next interest calculation is done on both the principal and the interest.

Interest can be compounded at any interval, but the most common compounding intervals are

- Annual: once per year.
- Quarterly: four times per year
- Monthly: 12 times per year
- Weekly: 52 times per year
- Daily: 365 times per year

Similarities and differences

While both types of interest will grow your money over time, there is a big difference between the two. Specifically, simple interest is only paid on principal, while compound interest is paid on the principal plus all of the interest that has previously been earned.

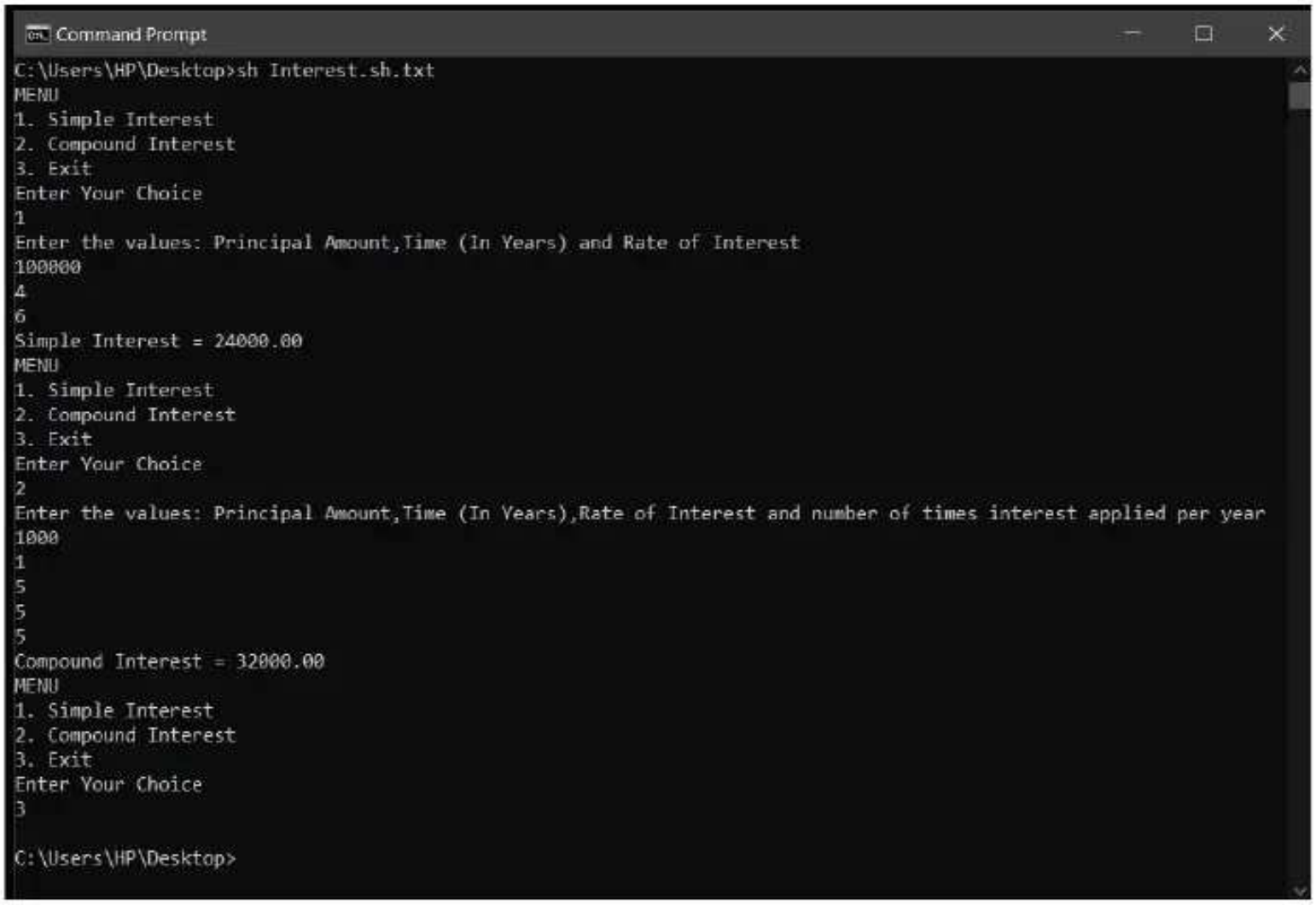
As an investor or depositor, you definitely want to earn compound interest, as it adds up greater over time. In the example of the $1,000 five-year CD at 4%, a simple interest calculation would produce $200, $21 less than the monthly compounding.

PROGRAM TO CALCULATE SIMPLE AND COMPOUND INTEREST USING SHELL SCRIPT

```
while [ 1 ]
do
   echo "MENU"
   echo "1. Simple Interest"
   echo "2. Compound Interest"
   echo "3. Exit"
   echo "Enter Your Choice"
   read choice
   case $choice in
1) echo "Enter the values: Principal Amount,Time (In Years) and Rate of Interest"
read p
read t
read r
si=`echo "scale=2;( $p * $t * $r ) / 100" | bc`
echo "Simple Interest = $si" ;;
2) echo "Enter the values: Principal Amount,Time (In Years),Rate of Interest and number of times interest applied per year"
read p
read t
read r
```

```
read n

power=`expr $n \* $t`

echo $power

ci=`echo "scale=2;  $p * ( 1 +  $r / $n  ) ^ $power" | bc`

echo "Compound Interest = $ci" ;;

3) exit;;

*)   echo "Invalid Choice";;

esac

done
```

Calculator Input And Output



Chapter 4

Conclusion

The accumulation function theorem can be derived from the practical example and proved. When a term is less than a period, the accumulated value of simple interest is more than that of compound interest; When a term is equal to a period, the accumulated value of simple interest is the same as that of compound interest.

Compound Interest gives a high return as compared to Simple Interest. In Simple Interest, the principal remains constant while in the case of Compound Interest the Principal changes due to the effect of compounding. The growth rate of Simple Interest is lower than the Compound Interest.

Chapter 5

References

https://docs.cs.cf.ac.uk/notes/linux-shell-commands/

https://prepinsta.com/simple-compound-interest/formulas/

https://www.examsbook.com/simple-and-compound-interest-formula