**Project-Title**

Basic interest calculator that computes both Simple Interest (SI) and Compound Interest (CI)

**Project Report**

Object Oriented Programming using Java (PGCSA111)

# Master of Computer Application

**PROJECT GUIDE:**

**Mr. Narayan Vyas (Internal)**

**SUBMITTED BY:**

**HEMANT MAURYA** (24CSA3BC006)
**JANVEE SHARMA** (24CSA3BC102)
**JITENDRA KUMAR** (24CSA3BC038)
**JYOTHI MOURYA** (24CSA3BC058)
**KAJAL PAREEK** (24CSA3BC063)

**April,2025**

**VIVEKANANDA GLOBAL UNIVERSITY**

# ACKNOWLEDGEMENT

I have taken this opportunity to express my gratitude and humble regards to the Vivekananda Global University to provide an opportunity to present a project on the "------------------" which is a Java programming-based project.

I would also be thankful to my project guide **Mr. Narayan Vyas** to help me in the completion of my project and the documentation. I have taken efforts in this project but the success of this project would not be possible without their support and encouragement.

I would like to thanks our Dean sir "Dr.R.C. Tripathi" to help us in providing all the necessary books and other stuffs as and when required. I show my gratitude to the authors whose books has been proved as the guide in the completion of my project I am also thankful to my classmates and friends who have encouraged me in the course of completion of the project.

Thanks

**HEMANT MAURYA** (24CSA3BC006)
**JANVEE SHARMA** (24CSA3BC102)
**JITENDRA KUMAR** (24CSA3BC038)
**JYOTHI MOURYA** (24CSA3BC058)
**KAJAL PAREEK** (24CSA3BC063)

**Place : Jaipur**

**Date: 07/04/2025**

# DECLARATION

We hereby declare that this Project Report titled "**Basic interest calculator that computes both Simple Interest (SI) and Compound Interest (CI)**" submitted by us and approved by our project guide, to the Vivekananda Global University, Jaipur is a bonafide work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

**Project Group**

| | |
|---|---|
| **Student Name:** | **HEMANT MAURYA** (24CSA3BC006)<br>**JANVEE SHARMA** (24CSA3BC102)<br>**JITENDRA KUMAR** (24CSA3BC038)<br>**JYOTHI MOURYA** (24CSA3BC058)<br>**KAJAL PAREEK** (24CSA3BC063) |
| **Project Guide:** | **Mr. Narayan Vyas** |

# Table of Contents

## 1. Introduction

This section provides a brief overview of the topic. You explain what interest is, why it's important in financial systems, and how it affects savings, loans, and investments. Mention that your project or program focuses on calculating **Simple Interest (SI)** and **Compound Interest (CI)** using a programming language.

- **Example:**
  Interest is the cost of borrowing money or the reward for saving it. It plays a critical role in personal finance and business transactions. This project demonstrates how to compute simple and compound interest through a computer program.

## 2. Objectives

Clearly state what the project aims to achieve. Objectives should be concise and focused.

- **Example Objectives:**
  - To understand and implement formulas for SI and CI.
  - To develop a program that calculates interest based on user input.
  - To compare SI and CI over time.
  - To demonstrate the impact of different parameters (rate, time, principal) on interest.

## 3. Interest Concepts

This section explains the theoretical background of interest, split into two subtopics:

## 3.1 Simple Interest (SI)

- **Definition:** SI is calculated only on the principal amount.
- **Formula**
- SI=P×R×T100\text{SI}=\frac{P\timesR\timesT}{100}SI=1 0P×R×T
  where:
- PPP = Principal
- RRR = Rate of Interest per annum
- TTT = Time in years
- **Use case:** Often used in short-term loans or basic savings schemes.

## 3.2 Compound Interest (CI)

- **Definition:** CI is calculated on the principal and also on the interest earned previously.
- **Formula:**
  CI=P×(1+R100)T−P\text{CI} = P \times \left(1 + \frac{R}{100}\right)^T - PCI=P×(1+100R)T−P
- **Use case:** Used in investments, recurring deposits, and long-term savings.
- CI grows faster than SI due to interest on interest.

## 4. Program Overview]

Describe the software/program you created to calculate SI and CI.

### 4.1 Language Used

Mention the programming language used (e.g., Python, C++, Java).

**Example:**
The program was written in **Python** because of its simplicity, readability, and support for mathematical operations.

## 4.2 Features

Highlight the key functionalities of your program.
**Example Features:**

- User-friendly input prompts for Principal, Rate, and Time.
- Separate outputs for SI and CI.
- Option to compare SI and CI.
- Clear and formatted output display.

## 4.3 Code Explanation

Provide a brief walkthrough of your code logic:

- Input section (user provides P, R, T)
- Calculation using formulas
- Displaying results
- Error handling (optional)

Optionally, include code snippets and explain them line by line.

**CODE :-**

**Part 1.**

- import javax.swing.*;
- import java.awt.*;
- import java.awt.event.ActionEvent;
- import java.awt.event.ActionListener;

- public class GUI extends JFrame {
- private JTextField principalField, rateField, timeField,compoundField;
- private JLabel resultLabel;
- public GUI() {
- setTitle("Interest Calculator");
- setSize(400, 300);

- setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
- setLayout(new GridLayout(6, 2, 10, 10));

// Creating UI Components

- add(new JLabel("Principal Amount:"));
- principalField = new JTextField();
- add(principalField);

- add(new JLabel("Rate of Interest (% per year):"));
- rateField = new JTextField();
- add(rateField);

- add(new JLabel("Time (years):"));
- timeField = new JTextField();
- add(timeField);

- add(new JLabel("Compounded per Year (Enter 1 for Simple Interest):"));
- compoundField = new JTextField();

- add(compoundField);

- JButton calculateButton = new JButton("Calculate");
- add(calculateButton);

- resultLabel = new JLabel("Result will be shown here.");
- add(resultLabel);

- // Action Listener for Button Click
- calculateButton.addActionListener(new ActionListener() {
    - @Override
    - public void actionPerformed(ActionEvent e) {
        - calculateInterest();
    - }
- });

- setVisible(true);
- }
- private void calculateInterest() {
- try {
    - double principal = Double.parseDouble(principalField.getText());
    - double rate = Double.parseDouble(rateField.getText());
    - double time = Double.parseDouble(timeField.getText());
    - int n = Integer.parseInt(compoundField.getText());

    - double simpleInterest = (principal * rate * time) / 100;
    - double compoundInterest = principal * Math.pow((1 + (rate / (100 * n))), n * time) - principal;

- resultLabel.setText("<html>Simple Interest: " + simpleInterest + "<br>Compound Interest: " + compoundInterest + "</html>");
- } catch (Exception ex) {
  - resultLabel.setText("Invalid input. Please enter numeric values.");
- }
- }

- public static void main(String[] args) {
- new GUI();
- }
- }

**Part 2.**

- import java.util.Scanner;
- public class TEXT{
- public static void main(String[] args) {
- Scanner scanner = new Scanner(System.in);

// Get user input

- System.out.print("Enter Principal Amount: ");
- double principal = scanner.nextDouble();
- System.out.print("Enter Rate of Interest (% per year): ");
- double rate = scanner.nextDouble();

- System.out.print("Enter Time (years): ");
- double time = scanner.nextDouble();
- System.out.print("Enter Number of Times Interest is Compounded per Year: ");
- int n = scanner.nextInt();

Calculate Simple Interest

- double simpleInterest = calculateSimpleInterest(principal, rate, time);
- System.out.println("Simple Interest: " + simpleInterest);

Calculate Compound Interest

- double compoundInterest = calculateCompoundInterest(principal, rate, time, n);
- System.out.println("Compound Interest: " + compoundInterest);
- scanner.close();
- }

Method to calculate Simple Interest

- public static double calculates Simple Interest (double principal, double rate, double time) {
- return (principal * rate * time) / 100;
- }

Method to calculate Compound Interest

- public static double calculateCompoundInterest(double principal, double rate, double time, int n) {
- return principal * Math.pow((1 + (rate / (100 * n))), n * time) - principal;
- }
- }

## 5. Applications

Describe real-world scenarios where SI and CI calculations are important:

- **Banking:** Calculating interest on loans and savings.
- **Education:** Student loans or savings plans.
- **Business:** Investment returns, project evaluations.
- **Personal Finance:** Home loans, car loans, credit card interest.

## 6. Future Enhancements

Mention possible upgrades to your project in the future.

**Examples:**

- Add GUI using Tkinter (Python) or JavaFX (Java).
- Include monthly/yearly compounding options.
- Store user data and interest history.
- Create a mobile or web app version.

## 7. Conclusion

The Basic Interest Calculator project, which allows users to calculate both Simple Interest and Compound Interest, is a practical and educational tool that serves both as a functional application and as an excellent learning resource for Java developers. By implementing this project, developers gain hands-on experience in Java programming, event-driven GUI development, and the application of mathematical formulas for financial calculations.

The application is built using Java Swing for the user interface, making it cross-platform compatible and accessible on various operating systems without the need for additional software. The project also provides an interactive interface that can be easily customized to suit user needs, with features such as input validation, error handling, and clear output display.

From a financial perspective, the calculator helps users quickly and accurately compute interest on investments or loans, making it a useful tool for students, professionals, or anyone interested in managing their finances effectively. Furthermore, it serves as an important stepping stone for developers to enhance their understanding of Java and how to create interactive applications.

**Example:**
This project successfully demonstrates how to calculate and compare Simple and Compound Interest using a basic program. It reinforces mathematical understanding and shows practical financial applications.