PART -B

2. Write a program menu driven to create a BankAccount class. class should support the following methods for i) Deposit ii) Withdraw iii) GetBalanace . Create a subclass SavingsAccount class that behaves just like a BankAccount, but also has an interest rate and a method that increases the balance by the appropriate amount of interest.

Code:

```
class BankAccount:
    def __init__(self):
        self.balance = 0


    def Deposit(self, amount):
        self.balance += amount
        print("Deposit amount:", amount)


    def Withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
            print("Withdrawn Amount:", amount)
        else:
            print("Insufficient Balance")


    def GetBalance(self):
        print("Current Balance:", self.balance)



class SavingAccount(BankAccount):
    def __init__(self, interest_rate):
        super().__init__()
```

```
        self.interest_rate = interest_rate

    def AddInterest(self):
        interest = self.balance * self.interest_rate / 100
        self.balance += interest
        print("Added interest amount:", interest)



ch = 1
while ch:
    if ch != 1:
        break
    account_type = input("Enter account type:\n 1. Bank Account \n 2. Saving account:")
    if account_type == "1":
        account = BankAccount()
    elif account_type == "2":
        interest_rate = float(input("Enter interest rate: "))
        account = SavingAccount(interest_rate)
    else:
        print("Invalid account type")
        exit()

    while True:
        print("\n Choose an operation ")
        print("1. Deposit")
        print("2. Withdraw")
        print("3. Get Balance")
```

```
if isinstance(account, SavingAccount):
    print("4. Add interest")

print("5. Exit")
choice = int(input("Enter choice: "))

if choice == 1:
    amount = float(input("Enter the deposit amount: "))
    account.Deposit(amount)
elif choice == 2:
    amount = float(input("Enter the withdrawal amount: "))
    account.Withdraw(amount)
elif choice == 3:
    account.GetBalance()
elif choice == 4 and isinstance(account, SavingAccount):
    account.AddInterest()
elif choice == 5:
    break
else:
    print("Invalid choice")

ch = int(input("Do you want to continue? Press 1 for yes, 0 for no: "))
```

Output:

```
Enter account type:
 1. Bank Account
 2. Saving account:1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 3
Current Balance: 0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 1
Enter the deposit amount: 2000
Deposit amount: 2000.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 3
Current Balance: 2000.0
Do you want to continue? Press 1 for yes, 0 for no: 1
```

```
 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 2
Enter the withdrawal amount: 1500
Withdrawn Amount: 1500.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 3
Current Balance: 500.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 2
Enter the withdrawal amount: 500
Withdrawn Amount: 500.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 3
Current Balance: 0.0
Do you want to continue? Press 1 for yes, 0 for no: 1
```

```
 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 2
Enter the withdrawal amount: 100
Insufficient Balance
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
5. Exit
Enter choice: 5
Enter account type:
 1. Bank Account
 2. Saving account:2
Enter interest rate: 4

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 3
Current Balance: 0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 1
Enter the deposit amount: 1500
Deposit amount: 1500.0
```

```
 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 3
Current Balance: 1560.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 2
Enter the withdrawal amount: 1500
Withdrawn Amount: 1500.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 3
Current Balance: 60.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 2
```

```
Enter the withdrawal amount: 60
Withdrawn Amount: 60.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 3
Current Balance: 0.0
Do you want to continue? Press 1 for yes, 0 for no: 1

 Choose an operation
1. Deposit
2. Withdraw
3. Get Balance
4. Add interest
5. Exit
Enter choice: 2
Enter the withdrawal amount: 2340
Insufficient Balance
Do you want to continue? Press 1 for yes, 0 for no: 0
```

3. Create a GUI to input Principal amount, rate of interest and number of years, Calculate Compound interest. When button submit is pressed Compound interest should be displayed in a textbox. When clear button is pressed all contents should be cleared.

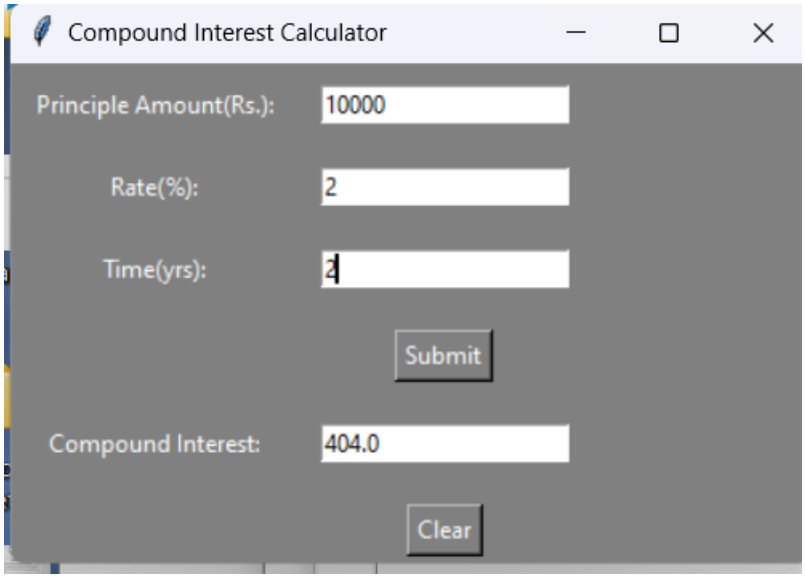
Code:

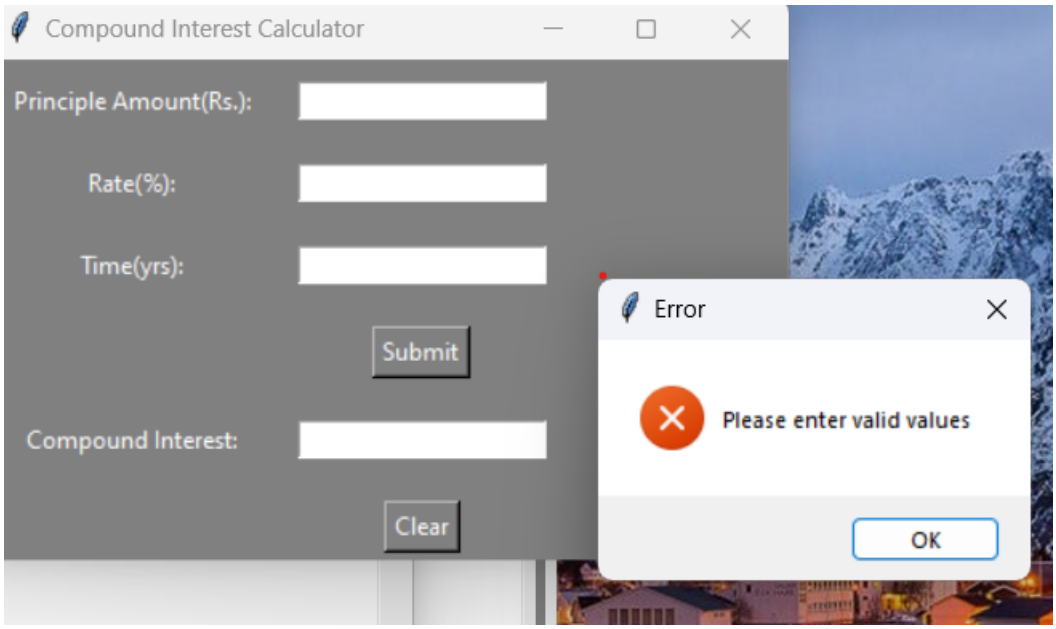
```
from tkinter import *

from tkinter import messagebox


def clear_all():

    principle_field.delete(0, END)

    rate_field.delete(0, END)

    time_field.delete(0, END)

    compound_field.delete(0, END)


def calculate_ci():

    try:

        principle = int(principle_field.get())

        rate = float(rate_field.get())

        time = int(time_field.get())

        CI = principle * (pow((1 + rate / 100), time)) - principle

        compound_field.insert(10, CI)

    except ValueError:

        messagebox.showerror("Error", "Please enter valid values")


if __name__ == "__main__":

    root = Tk()

    root.configure(background='grey')

    root.geometry("400x250")  # Corrected typo in the geometry dimensions
```

```
root.title('Compound Interest Calculator')

label1 = Label(root, text="Principle Amount(Rs.):", fg="white", bg="grey")
label2 = Label(root, text="Rate(%):", fg="white", bg="grey")
label3 = Label(root, text="Time(yrs):", fg="white", bg="grey")
label4 = Label(root, text="Compound Interest:", fg="white", bg="grey")

label1.grid(row=1, column=0, padx=10, pady=10)
label2.grid(row=2, column=0, padx=10, pady=10)
label3.grid(row=3, column=0, padx=10, pady=10)
label4.grid(row=5, column=0, padx=10, pady=10)

principle_field = Entry(root)
rate_field = Entry(root)
time_field = Entry(root)
compound_field = Entry(root)

principle_field.grid(row=1, column=1, padx=10, pady=10)
rate_field.grid(row=2, column=1, padx=10, pady=10)
time_field.grid(row=3, column=1, padx=10, pady=10)  # Corrected row number
compound_field.grid(row=5, column=1, padx=10, pady=10)

btn1 = Button(root, text="Submit", bg="grey", fg="white", command=calculate_ci)
btn2 = Button(root, text="Clear", bg="grey", fg="white", command=clear_all)

btn1.grid(row=4, column=1, pady=10)
btn2.grid(row=6, column=1, pady=10)
```

```
root.mainloop()
```

Output:





4. Write a GUI program to implement Simple Calculator

Code:

```
from tkinter import *


expression=" "


def press(num):
    global expression
    expression=expression+str(num)
    equation.set(expression)


def equalpress():
    try:
        global expression
        total=str(eval(expression))
        equation.set(total)
        expression=" "
    except:
        equation.set("error")
        expression=" "


def clear():
    global expression
    expression=" "
    equation.set(" ")


def create_button(label,row,column):
```

```
    if label=="=":

button=Button(gui,text='=',fg='black',bg='white',command=equalpress,height=1,width=7)
        button.grid(row=row,column=column)
    elif label=="C":
        button=Button(gui,text='clear',fg='black',bg='white',command=clear,height=1,width=7)
        button.grid(row=row,column=column)
    else:

button=Button(gui,text=label,fg='black',bg='white',command=lambda:press(label),height=1,
width=7)
        button.grid(row=row,column=column)
        expression=""



if __name__=="__main__":
    gui=Tk()
    gui.configure(background="grey")
    gui.title("Simple Calculator")
    gui.title("270x150")
    equation=StringVar()
    expression_field=Entry(gui,textvariable=equation)
    expression_field.grid(columnspan=4,ipadx=70)
    button_labels=['1','2','3','+','4','5','6','-','7','8','9','*','0','.','C','/','=']
    row=2
    column=0
    for label in button_labels:
        create_button(label,row,column)
```

```
        column+=1
        if column>3:
            column=0
            row+=1
gui.mainloop()
```