INSTITUTE OF TECHNOLOGY AND MANAGEMENT
SKILLS UNIVERSITY,
KHARGHAR, NAVI MUMBAI

PYTHON PROGRAMMING LAB

Prepared by:

Name of Student: Shikha singh________________________

Roll No: 25____________

Batch: 2023-27

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

xp. No	List of Experiment
	1.1 Write a program to compute Simple Interest.
	1.2 Write a program to perform arithmetic, Relational operators.
	1.3 Write a program to find whether a given no is even & odd.
	1.4 Write a program to print first n natural number & their sum.
	1.5 Write a program to determine whether the character entered is a Vowel or not .
	1.6 Write a program to find whether given number is an Armstrong Number.
	1.7 Write a program using for loop to calculate factorial of a No.
	1.8 Write a program to print the following pattern
	i) * * * * * * * * * * * * * * *
	ii) 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5

	iii)
	        *
	      * * *
	    * * * * *
	  * * * * * * *
	* * * * * * * * *

	2.1 Write a program that define the list of defines the list of define countries that are in BRICS.
	2.2 Write a program to traverse a list in reverse order. 1.By using Reverse method. 2.By using slicing
	2.3 Write a program that scans the email address and forms a tuple of username and domain.
	2.4 Write a program to create a list of tuples from given list having number and add its cube in tuple. i/p: c= [2,3,4,5,6,7,8,9]
	2.5 Write a program to compare two dictionaries in Python? (By using == operator)
	2.6 Write a program that creates dictionary of cube of odd numbers in the range.
	2.7 Write a program for various list slicing operation. a= [10,20,30,40,50,60,70,80,90,100] i. Print Complete list ii. Print 4th element of list iii. Print list from0th to 4th index. iv. Print list -7th to 3rd element v. Appending an element to list. vi. Sorting the element of list. vii. Popping an element. viii. Removing Specified element. ix. Entering an element at specified index. x. Counting the occurrence of a specified element. xi. Extending list. xii. Reversing the list.
	3.1 Write a program to extend a list in python by using given approach. i. By using + operator. ii. By using Append ()

	iii. By using extend ()
	3.2 Write a program to add two matrices.
	3.3 Write a Python function that takes a list and returns a new list with distinct elements from the first list.
	3.4 Write a program to Check whether a number is perfect or not.
	3.5 Write a Python function that accepts a string and counts the number of upper- and lower-case letters. string_test= 'Today is My Best Day'
	4.1 Write a program to Create Employee Class & add methods to get employee details & print.
	4.2 Write a program to take input as name, email & age from user using combination of keywords argument and positional arguments (*args and**kwargs) using function,
	4.3 Write a program to admit the students in the different Departments(pgdm/btech) and count the students. (Class, Object and Constructor).
	4.4 Write a program that has a class store which keeps the record of code and price of product display the menu of all product and prompt to enter the quantity of each item required and finally generate the bill and display the total amount.
	4.5 Write a program to take input from user for addition of two numbers using (single inheritance).
	4.6 Write a program to create two base classes LU and ITM and one derived class. (Multiple inheritance).
	4.7 Write a program to implement Multilevel inheritance, Grandfather□Father-□Child to show property inheritance from grandfather to child.
	4.8 Write a program Design the Library catalogue system using inheritance take base class (library item) and derived class (Book, DVD & Journal) Each derived

	class should have unique attribute and methods and system should support Check in and check out the system. (Using Inheritance and Method overriding)
	5.1 Write a program to create my_module for addition of two numbers and import it in main script.
	5.2 Write a program to create the Bank Module to perform the operations such as Check the Balance, withdraw and deposit the money in bank account and import the module in main file.
	5.3 Write a program to create a package with name cars and add different modules (such as BMW, AUDI, NISSAN) having classes and functionality and import them in main file cars.
	6.1 Write a program to implement Multithreading. Printing “Hello” with one thread & printing “Hi” with another thread.
.	7.1 Write a program to use ‘whether API’ and print temperature of any city, also print the sunrise and sunset times for the same humidity of that area.
	7.2 Write a program to use the ‘API’ of crypto currency.

Name of Student: _Shikha singh________________

Roll Number: _25________________

Experiment No: 5.2

Title:

5.2 Write a program to create the Bank Module to perform the operations such as

Check the Balance, withdraw and deposit the money in bank account and import the module in main file.

Theory:

- **Class Definition:**
  - **The Bank Module defines a class named Bank. A class is a blueprint for creating objects, and objects are instances of a class.**
- **Constructor (__init__ method):**
  - **The __init__ method is a special method called a constructor. It initializes the state of the object when an instance of the class is created. In this case, it initializes the initial balance.**
- **Class Methods:**
  - **The class contains methods such as check_balance, withdraw, and deposit that define the operations that can be performed on a bank account.**

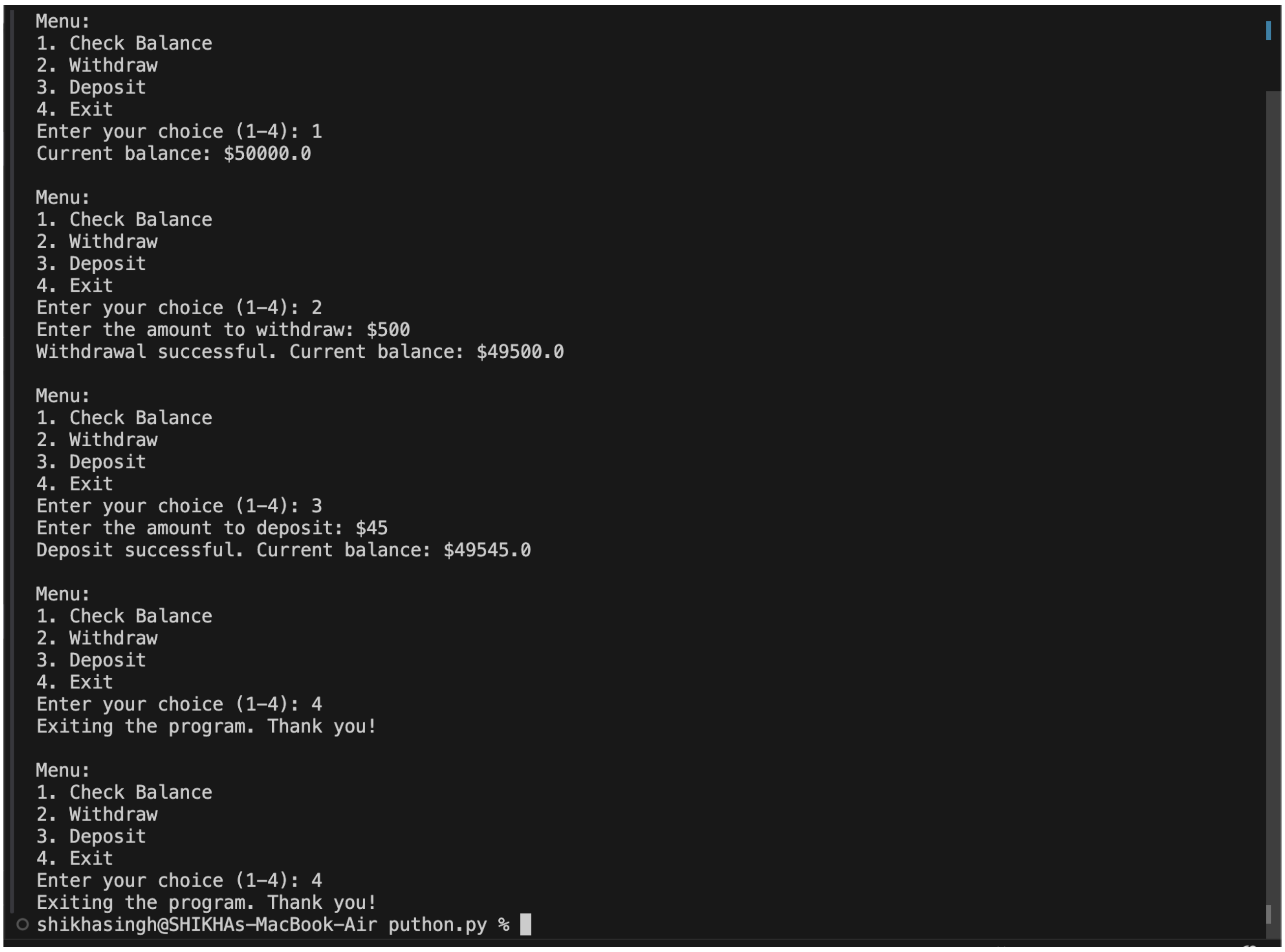
Main Script:
The main script interacts with the Bank Module and provides a user interface. Here are some key concepts:

- **Importing Modules:**
  - **The from bank_module import Bank statement is used to import the Bank class from the bank_module module. This allows the main script to use the functionality defined in the module.**
- **User Input Handling:**
  - **The program uses a function get_valid_input to handle user input. This function ensures that the user enters a valid numerical value, handling exceptions for non-numeric inputs.**
- **Menu-driven Interface:**
  - **The main script presents a menu to the user, allowing them to choose between different operations (checking balance, withdrawing, depositing, or exiting). This is achieved using a loop that repeatedly displays the menu and performs the selected operation.**
- **Interactive User Experience:**
  - **The program provides an interactive user experience by allowing the user to perform multiple operations until they choose to exit. This is achieved using a while loop that continues until the user selects the exit option.**

Code:

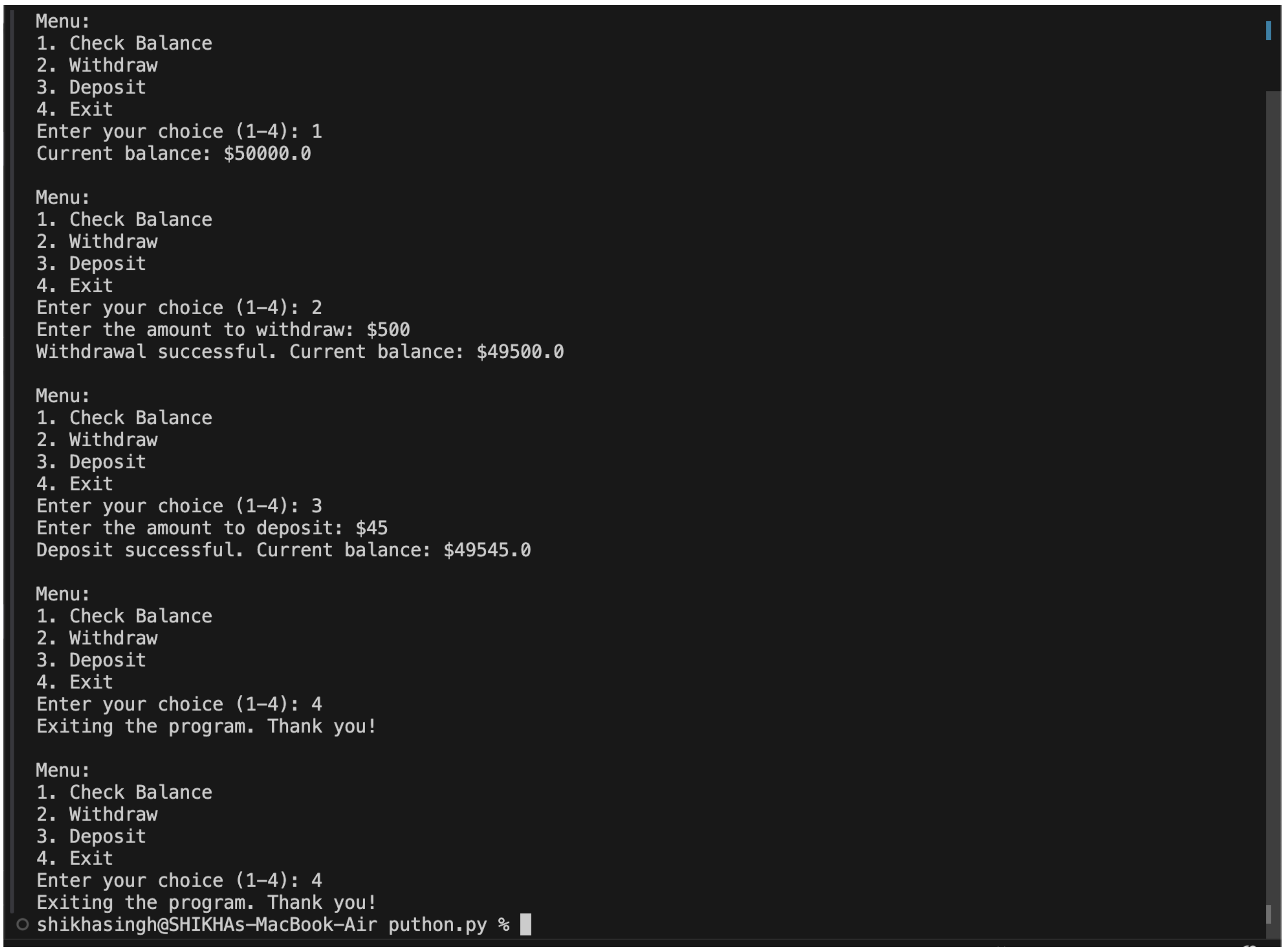
```
# bank_module.py
class Bank:
    def __init__(self, initial_balance):
        self.balance = initial_balance
    def check_balance(self):
        return self.balance
    def withdraw(self, amount):
        if amount > 0 and amount <= self.balance:
            self.balance -= amount
            print(f"Withdrawal successful. Current balance: ${self.balance}")
        else:
            print("Invalid withdrawal amount or insufficient funds.")
    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposit successful. Current balance: ${self.balance}")
        else:
            print("Invalid deposit amount.")
# main_script.py
from bank_module import Bank
def display_menu():
    print("\nMenu:")
    print("1. Check Balance")
    print("2. Withdraw")
    print("3. Deposit")
    print("4. Exit")
    while True:
        try:
            choice = int(input("Enter your choice (1-4): "))
            if 1 <= choice <= 4:
                return choice
            else:
                print("Invalid choice. Please enter a number between 1 and 4.")
        except ValueError:
            print("Invalid input. Please enter a valid number.")
# Initialize the initial balance to 50000
initial_balance = 50000.0
my_bank = Bank(initial_balance)
while True:
    user_choice = display_menu()
    if user_choice == 1:
        # Check Balance
        print(f"Current balance: ${my_bank.check_balance()}")
    elif user_choice == 2:
        # Withdraw
        try:
            withdraw_amount = float(input("Enter the amount to withdraw: $"))
            my_bank.withdraw(withdraw_amount)
        except ValueError:
            print("Invalid input. Please enter a valid number.")
    elif user_choice == 3:
        # Deposit
        try:
            deposit_amount = float(input("Enter the amount to deposit: $"))
            my_bank.deposit(deposit_amount)
        except ValueError:
            print("Invalid input. Please enter a valid number.")
    elif user_choice == 4:
        # Exit
        print("Exiting the program. Thank you!")
        break
```

Output: (screenshot)



```
Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 1
Current balance: $50000.0

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 2
Enter the amount to withdraw: $500
Withdrawal successful. Current balance: $49500.0

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 3
Enter the amount to deposit: $45
Deposit successful. Current balance: $49545.0

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 4
Exiting the program. Thank you!

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 4
Exiting the program. Thank you!
shikhasingh@SHIKHAs-MacBook-Air puthon.py %
```

Test Case: Any two (screenshot)



```
Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 1
Current balance: $50000.0

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 2
Enter the amount to withdraw: $500
Withdrawal successful. Current balance: $49500.0

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 3
Enter the amount to deposit: $45
Deposit successful. Current balance: $49545.0

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 4
Exiting the program. Thank you!

Menu:
1. Check Balance
2. Withdraw
3. Deposit
4. Exit
Enter your choice (1-4): 4
Exiting the program. Thank you!
shikhasingh@SHIKHAs-MacBook-Air puthon.py %
```

Conclusion:

The modular structure of the program, with a separate Bank Module, promotes code organization, reusability, and maintainability. The main script serves as an interface to interact with the Bank Module, providing a user-friendly way to perform banking operations.