BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS), BAGALKOTE-587102

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS

NAME : SUHAS.R.HATTI

PRIYA.M.*MALAGHAN*

CLASS : 3RD SEM " B "

USN : 2BA21EC108 *ROLL NO : 049,030*

2BA21EC070

SUBJECT :-- HIGHER PROGRAMMING PARADIGM

COURSE CODE :-- 21UEC308C

FACULTY :--PROF.M.C.ARALIMARAD

SIGN

BASAVESHWAR ENGINEERING COLLEGE (AUTONOMOUS), BAGALKOTE-587102

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS

CERTIFICATE

This is to certify that project entitled "**CREATING A BANK ACCOUNT USING PYTHON**" a bonafied work of**, Mr.SUHAS.HATTI , Miss.PRIYA.M.MALAGHAN**. The report satisfies the academic requirements with respect to project work prescribed for 3th semester during the academic year 2022-2023. It is certified that all corrections/suggestions indicated assessement of the project have been satisfied.

PROJECT GUIDE :	HEAD OF THE DEPARTMENT :
PROF.M.C.ARALIMARAD	DR.SHRIDHAR KUNTOJI

SIGNATURE WITH DATE :

INTRODUCTION :--

Object oriented programming is the principle of design and development of programs using modular approach. Object oriented programming approach provides advantages in creation and development of software for real life applications. The advantage is that small modules of program can be developed in shorter span of time and these modules can be shared by a number of applications. The basic element of object oriented programming is the data. The programs are built by combining data and functions that operate on the data. In this chapter we learn about the basic concepts, advantages and terminologies used in Object oriented programming. Some of the object oriented programming languages are C++, Java, C# and so on.

The object oriented programming methods use data as the main element in the program. The data is tied to the function that operates on the data and the other functions cannot modify the data tied to a given function. Thus in object oriented programming, a problem is decomposed into a number of components called objects. An object is a collection of set of data known as member data and the functions that operate on these data a known as member functions or Methods. The member data are encapsulated in an object and then can be accessed or modified only by the member functions. An object can be accessed only if permitted by other member functions. Various objects of a program can interact with each other by sending messages.

Object oriented programming methods modularize a program by creating memory area for data and member functions (methods) together as a single entity. All objects are created according to the specifications of the entity defined. Object is the basic unit of OOP. To *design OO Model, first a set of classes are defined. A class is a Template from which objects are created. The Template of a class specifies the data, member functions and their attributes.*

BASICS CONCEPTS OF OBJECT ORIENTED PROGRAMING:--

1.Objects

2.Classes

3. Data Abstraction

4.Data Encapsulation

5.Inheritance

6.Overloading

7.Polymorphsim

8.Dynamic Binding

9.Message Passing

OBJETS:--

Objects are basic building blocks for designing programs. An object may represent a person, place or a table of data. An object is a collection of data members and associated member functions. Each object is identified by a unique name. Every object must be a member of a particular class.

Ex: Apple, orange, mango are the objects of class fruit.

CLASSES :--

The objects can contain data and code to manipulate the data. The objects can be made user defined data types with the help of a class. Therefore objectsare variables of the type class. A class is a way of *grouping objects having similar characteristics. Once a class is defined, any number of objects of that class are created.*

DATA ABSTRACTION :--

Data abstraction permits the user to use an object without knowing its internal working. Abstraction refers to the process of representing essential features without including background details or explanations. Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight and cost, and functions to operate on these attributes.

DATA ENCAPSULATION:--

Data encapsulation combines data and functions into a single unit called class. Data encapsulation will prevent direct access to data. The data can be accessed only through methods (function) present inside the class. The data cannot be modified by an external non- member function of a class. Data encapsulation enables data hiding or information hiding.

INHERITANCE:--

In OOP, the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it. Thus the process of forming a new class from an existing class is known as Inheritance. The objects of one class acquire the properties of another class through inheritance. The existing class is known as base class. The new class is known as derived class.

OVERLOADING :--

Function overloading means two or more functions have same name,but differ in the number of arguments or data type of arguments. Therefore it is said that (function name) is overloaded. Function overloading therefore is the pro- cess of defining same function name to carry out similar types of activities with various data items.

POLYMORPHISM :--

Polymorphism is a feature of object oriented programming where a function can take multiple forms based on the type of arguments, number of arguments and data type of return value.

DYNAMIC BINDING:--

Binding is the process of connecting one program to another. Dynamic binding means code associated with a procedure call is known only at the time of program execution routine.

MESSAGE PASSING :--

In OOP, processing is done by sending messages to objects. A message for an object is request for execution of procedure. The request will involve a procedure (function) in the receiving object that generates desired results. Message passing involves specifying the name of object, the name of the function (message) and the information to be sent

PROGRAM :--

```
class Bank:
    def __init__(self):
        self.total_amount = 0
        self.name = ''

    def welcome(self):
        self.name = input('Welcome to your Bank Account. Please Enter your name : ')

    def print_current_balance(self):
        print('Your Current balance : {}'.format(self.total_amount))

    def deposit(self):
        self.total_amount += float(input('Hello {}, please enter amount to deposit : '.format(self.name)))
        self.print_current_balance()

    def withdraw(self):
        amount_to_withdraw = float(input('Enter amount to withdraw : '))

        if amount_to_withdraw > self.total_amount:
            print('Insufficient Balance !!')
        else:
            self.total_amount -= amount_to_withdraw

        self.print_current_balance()

if __name__=="__main__":
    bank = Bank()
    bank.welcome()

    while True:
        input_value = int(input('Enter 1 to see your balance,\n2 to deposit\n3 to withdraw\n'))

        if input_value == 1:
            bank.print_current_balance()
        elif input_value == 2:
            bank.deposit()
        elif input_value == 3:
            bank.withdraw()
        else:
            print('Please enter a valid input.')
```

OUTPUT:--

```
Welcome to your Bank Account. Please Enter your name :  ROME
Enter 1 to see your balance,
2 to deposit
3 to withdraw
2
Hello  ROME, please enter amount to deposit : 2000
Your Current balance : 2000.0
Enter 1 to see your balance,
2 to deposit
```

```
3 to withdraw

3

Enter amount to withdraw : 1000

Your Current balance : 1000.0

Enter 1 to see your balance,

2 to deposit

3 to withdraw

1

Your Current balance : 1000.0

Enter 1 to see your balance,

2 to deposit

3 to withdraw
```

PROGRAM EXPLATION :--

STEP 1 : Create a class `Bank_Acccount`. Then define a function using `__init__` with default argument `self`. This keyword is used in Python to initialize attributes of the class when an object of that class is created. This step is followed by initializing the balance as 0.

STEP 2: Now we have to create a function `deposit` such that the amount of money is taken by input using float and is then added to the balance. Then the amount deposited will be displayed using the print statement in the next line as shown in the code below:

STEP 3 : Just like step 2, we are going to create another function `withdraw` in which we are going to take float input for the amount to get withdraw. We are using an if condition here just to check if we have sufficient balance available to perform a withdrawal of any amount from the account. If the balance is not sufficient then our class will show "Insufficient balance".

STEP 4 : Now we are going to create our final function which is `display` function. It will display the final balance of the account after withdrawal and deposit.

STEP 5 : As a final step we are going to create an object of our class so that we can call all the functions with that class to execute our code.

CONCLUSION:---

1. *Got to know about oops*
2. *How to use the oops*
3. *Advantages and disadvantages of oops*
4. *Problem analysis*
5. *Banking system info*