

1. Bank Account class  
2. Interest class  
3. Interest class  
4. Interest class

5. Interest class  
6. Interest class  
7. Interest class

8. Interest class  
9. Interest class  
10. Interest class

11. Interest class  
12. Interest class  
13. Interest class

14. Interest class  
15. Interest class  
16. Interest class

17. Interest class  
18. Interest class  
19. Interest class

20. Interest class  
21. Interest class  
22. Interest class

23. Interest class  
24. Interest class  
25. Interest class

26. Interest class  
27. Interest class  
28. Interest class

29. Interest class  
30. Interest class  
31. Interest class

input: account number, initialize balance

set account Number and balance to input values

Method deposit:

input: amount

If amount is less than or equal to balance.

subtract amount from balance

else

Display "Insufficient Balance"

Main:

Create an object of Bank Account with account number and initial Balance

call deposit method to add money

call withdraw method to take out money

Q Aim: To write a Pseudo code for rectangle in constructor

Pseudo code

class rectangle

variables:

double length

double width

Constructor:

input: length, width

set length & width to input values

Method Calculate area:

return: length \* width

Main:

Create an object of rectangle with length and width  
call calculate area method to get and display area.

② Person

Pseudo code:

class person

Variables:

string name

id age

Constructor

input: name, age

Set name and age to input values

Method print PersonDetails

output: Display name & age

main:

Create a object of person with name and age

call print Person Details method to display the person's name and age

③ Overriding

Pseudo code:

Method add

input: two numbers

return: sum of the two numbers

method Subtract:

input: two numbers

return: sum of the two numbers

method subtract:

input: two numbers

return: difference between the two numbers

method multiply:

input: two numbers

return: product of two numbers

method divide:

input: two numbers

if denominator is not zero:

class  
calculator return: quotient

else

display: can't divide by zero

(Inherits  
calculator)

override method multiply:

input: two numbers

return: more complex calculations

main

create a object of calculator class

call add, sub, multi and division

Q) Overloading

class calculator:

method sum (int ... num):

Initialize total as 0

for each integer in num:

Add integer to total

return total

overloaded method sum (double num);

Initialize total as 0.0

for each double in number

add the double to total

return total

main:

Create an object of calculator class

Call sum method with a variable number of integers.

Call overloaded sum method with a variable number of doubles.

## Q Polymorphism

Pseudo code:

Abstract class Animal:

No implementation

Abstract method sleep();

No implementation

class Dog (Inherits Animal):

Implement method eat();

display: "Dog is eating"

implement Method sleep(),

display: "Dog is sleeping"

class cat (Inherits Animal):

Implement method sleep(),

display: "cat is sleeping"

main:

Create an object of Dog class

call eat and sleep methods

Create an object of class:

call eat and sleep methods

## ⑥ demonstrate polymorphism

Pseudo code:

Interface Drawable:

Method Draw():

no implement

class Circle:

Implement method draw:

display: "Drawing a Circle"

class Square:

implement method draw

display: "Drawing a Square"

main:

Create an object of circle class  
call draw method (circle)

Create an object of square class

call draw method (square)

19/08/24