

Lab#10  
Object oriented programming  
27/12/2022

**Task#01**

Create following classes:

**a. Shape :**

- i. Create 2 data members: width(protected),height(protected).
- ii. Create a constructor for variable initialization.
- iii. Create a virtual function area that simply multiply base and height in it..

**b. Rectangle:**

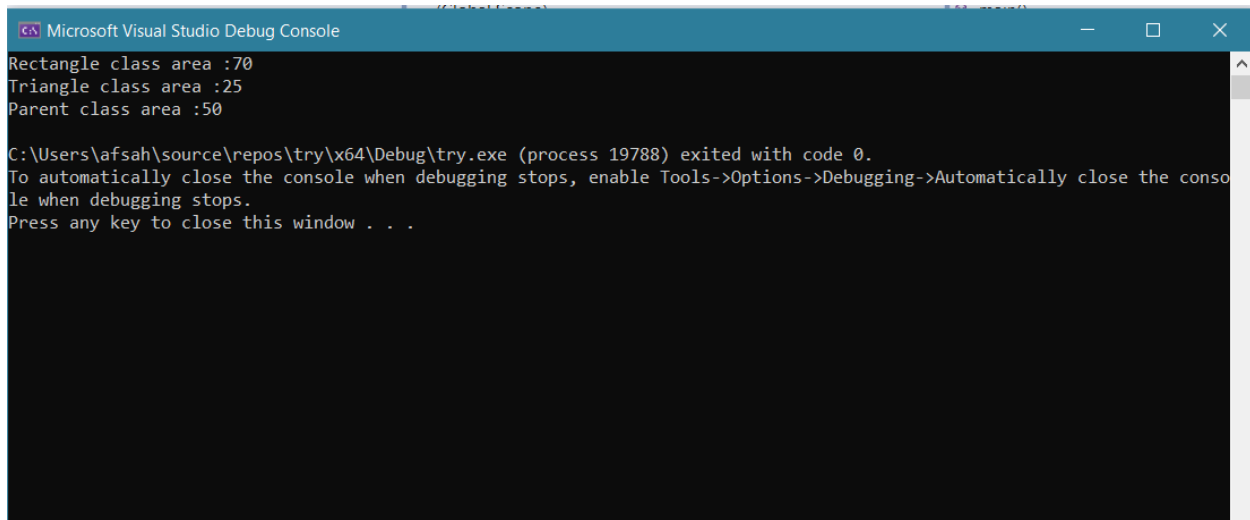
- i. Inherit from class shape.
- ii. Create a constructor for values initialization.
- iii. Override area function

**c. Triangle**

- i. Inherit from class shape.
- ii. Create a constructor for values initialization.
- iii. Override area function

**Main function:**

Display area of all 3 classes.



```
Microsoft Visual Studio Debug Console
Rectangle class area :70
Triangle class area :25
Parent class area :50

C:\Users\afsah\source\repos\try\x64\Debug\try.exe (process 19788) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

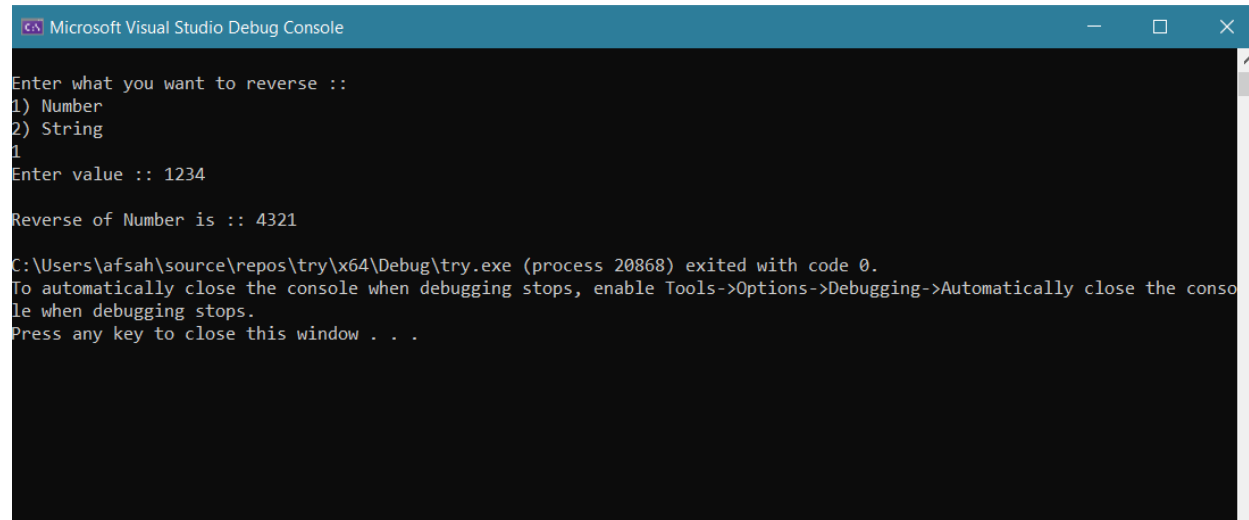
Lab#10  
Object oriented programming  
27/12/2022

**Task#02**

Write a C++ class Reverse that:

- a. Asks the user if he wants to reverse a number or c-string.
- b. Take input from the user.
- c. Create 2 functions named as reverse that reverses either number or c-string.

**Note:** The names of both functions must be the same.



```
Microsoft Visual Studio Debug Console
Enter what you want to reverse ::
1) Number
2) String
1
Enter value :: 1234
Reverse of Number is :: 4321
C:\Users\afsah\source\repos\try\x64\Debug\try.exe (process 20868) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Lab#10  
Object oriented programming  
27/12/2022

**Task#03**

Create following classes:

**a. Account:**

- i. Create data members: name(char), accno(int).
- ii. Member function: getAccountDetails(), displayDetails().

**b. Current\_account**

- i. Inherit from Account class.
- ii. Create data members: balance(float) and initialize with 500.
- iii. Member function:
  - 1. display() to display amount in account.
  - 2. deposit() to add an amount in balance.
  - 3. withdraw() to withdraw amount if balance is greater than 1000.
  - 4. Show amount after withdrawal and/deposit.
  - 5. Display error if Insufficient Balance.

**c. Saving\_account**

- i. Inherit from Account class.
- ii. Create data members: balance(float) and initialize with 500.
- iii. Member function:
  - 1. display() to display the amount in account.
  - 2. deposit() to add an amount in current balance with an interest rate added.

$$\text{Interest} = (\text{balance} * 2) / 100;$$

- 3. withdraw() to withdraw an amount if balance is greater than 1000.
- 4. Show amount after withdrawal and/deposit.
- 5. Display error if Insufficient Balance.

**d. In main:**

- i. Asks from user about account type i.e. current or saving account.
- ii. Asks user about action that he wants to perform. i.e.
  - 1) Deposit.
  - 2) Withdraw.
  - 3) Display Balance.
  - 4) Display with full Details.
  - 5) Exit.

And perform actions accordingly.

- iii. Create objects of derived classes using pointers of base class.

Lab#10  
Object oriented programming  
27/12/2022

```
C:\Users\afsah\source\repos\try\Debug\try.exe
Enter S for saving customer and C for current a/c customer : s
Enter Customer Name : john
Enter Account Number : 5236

Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 1

Enter amount to Deposit : 5000

Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 3

Balance : 5610
Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 2

Balance :- 5610
Enter amount to be withdraw : 600

Balance Amount After Withdraw: 5010
Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 3

Balance : 5010
Choose Your Choice
1) Deposit
2) Withdraw
```

Lab#10  
Object oriented programming  
27/12/2022

```
C:\Users\afsah\source\repos\try\x64\Debug\try.exe
Enter S for saving customer and C for current a/c customer : c
Enter Customer Name : ali
Enter Account Number : 4563

Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 1

Enter amount to Deposit : 6000

Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 2

Balance : 6500
Enter amount to be withdraw :500

Amount After Withdraw: 6000
Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 3

Balance :6000
Choose Your Choice
1) Deposit
2) Withdraw
3) Display Balance
4) Display with full Details
5) Exit
Enter Your choice: 4

Customer Name : ali
Account Number : 4563
```

Lab#10  
Object oriented programming  
27/12/2022

**Task#04**

There is a software house that is looking for a software that calculates increment on salaries of their employees on the basis of their experience. Chart below is defining increment criteria.

Write a program that takes name, address, phone number, experience and current salary from the user and displays his new salary.

Increment criteria	
Designation	Increment
Fresh	0%
Junior developer	20%
Senior developer	35%

- For each designation create a different class and inherit it from the developer class.
- Base class must have functions that take employee information ,a virtual function to calculate salary and a function that displays employee information.
- Create base class pointers to call derived classes functions.