

1. Aim: Define a **class** Shape whose attributes are radius, length and width calculate the perimeter of the rectangle and circle. Use **constructors and destructors**.

Software Used:

- **IDE:** Visual Studio Code
- **Terminal:** Windows Terminal
- **Shell:** zsh
- **Compiler:** gcc

Program:

```
1  #include<iostream>
2  using namespace std;
3
4  class Shape{
5      private:
6          float radius;
7          float length;
8          float breadth;
9
10     public:
11         // constructor for rectangle
12         Shape(float length, float breadth){
13             this->length = length;
14             this->breadth = breadth;
15
16             this->radius = 0;
17         }
18
19         // constructor for circle
20         Shape(float radius){
21             this->radius = radius;
22
23             this->length = 0;
24             this->breadth = 0;
25         }
26
27         // member function to calculate area of rectangle
28         float perimeter(){
29             if(radius == 0){
30                 return 2*(length + breadth);
31             }
32
33             return 2*3.14*radius;
34         }
35
36         ~Shape(){
37             cout << "Shape destroyed" << endl;
38         }
39 };
```

```
40
41
42 // driver
43 int main()
44 {
45     Shape circle(50), rectangle(10, 20);
46
47     cout << "Perimeter of Rectangle: " << rectangle.perimeter() << endl;
48     cout << "Perimeter of Circle: " << circle.perimeter() << endl << endl;
49
50     return 0;
51 }
52
53
```

Output:

```
🔗 /mnt/d/Ro/Do/A/S/3/S/0/Lab on 🐱 master
Perimeter of Rectangle: 60
Perimeter of Circle: 314

Shape destroyed
Shape destroyed
```

2. Aim: Create a class Person which includes: character array name of size 64, age in numeric, character array address of size 64, and total salary in real numbers (divide salary in different components, if required). Make an **array of objects** of class Person of size 10.

- (a) Write **inline** function that obtains the youngest and eldest age of a person from a class person using arrays.
- (b) Write a program to develop the salary slip and display result by using constructors.

Software Used:

- **IDE:** Visual Studio Code
- **Terminal:** Windows Terminal
- **Shell:** zsh
- **Compiler:** gcc

Program:

```
1  #include <iostream>
2  #include <cstring>
3
4  using namespace std;
5
6  class Person;
7
8  class Salary{
9      private:
10         float basic_salary;
11         float house_rent_allowance;
12         float medical_allowance;
13         float dearness_allowance;
14         float income_tax;
15         float gross_salary;
16
17         friend class Person;
18 };
19
20
21 class Person{
22     private:
23
24         // employee name
25         char name[64];
26         // employee age
27         int age;
28         // employee address
29         char address[64];
30         // employee salary
31         Salary salary;
32
33     public:
34
35         // employee count
36         static int employee_ages[100], employee_count;
```

```

37
38 // salary slip generator
39 void salary_slip(){
40     cout << "\n\n-----" << endl;
41     cout << "Name           : " << name << endl;
42     cout << "Age            : " << age << endl;
43     cout << "Address         : " << address << endl;
44     cout << "Basic Salary      : " << salary.basic_salary << endl;
45     cout << "House Rent Allowance : " << salary.house_rent_allowance << endl;
46     cout << "Medical Allowance   : " << salary.medical_allowance << endl;
47     cout << "Dearness Allowance  : " << salary.dearness_allowance << endl;
48     cout << "Income Tax         : " << salary.income_tax << endl;
49     cout << "Gross Salary       : " << salary.gross_salary << endl;
50 }
51
52
53 // default constructor
54 Person(){
55     strcpy(name, "No name");
56     age = 0;
57     strcpy(address, "No address");
58     salary.basic_salary = 0;
59     salary.house_rent_allowance = 0;
60     salary.medical_allowance = 0;
61     salary.dearness_allowance = 0;
62     salary.income_tax = 0;
63     salary.gross_salary = 0;
64
65     // increment employee count and add age to array
66     employee_ages[employee_count] = this->age;
67     employee_count++;
68 }
69
70 // parameterized constructor
71 Person(string name, int age, string address, float basic_salary,
72 float house_rent_allowance, float medical_allowance, float dearness_allowance,
73 float income_tax){
74
75     strcpy(this->name, name.c_str());
76     this->age = age;
77     strcpy(this->address, address.c_str());
78     this->salary.basic_salary = basic_salary;
79     this->salary.house_rent_allowance = house_rent_allowance;
80     this->salary.medical_allowance = medical_allowance;
81     this->salary.dearness_allowance = dearness_allowance;
82     this->salary.income_tax = income_tax;
83     this->salary.gross_salary = (basic_salary + house_rent_allowance +
84     medical_allowance + dearness_allowance - income_tax);
85
86     // increment employee count and add age to array
87     employee_ages[employee_count] = this->age;
88     employee_count++;
89
90     // display salary slip
91     salary_slip();
92 }
93

```

```

94         // obtain youngest employee age and eldest employee age from array
95         inline void youngest_eldest_employee_age(){
96             int youngest_employee_age = employee_ages[0], eldest_employee_age = employee_ages[0];
97             cout << endl;
98             for(int i = 0; i < employee_count; i++){
99                 if(employee_ages[i] < youngest_employee_age){
100                     youngest_employee_age = employee_ages[i];
101                 }
102                 if(employee_ages[i] > eldest_employee_age){
103                     eldest_employee_age = employee_ages[i];
104                 }
105             }
106             cout << "Youngest employee age: " << youngest_employee_age << endl;
107             cout << "Eldest employee age: " << eldest_employee_age << endl;
108         }
109     };
110 };
111
112 // static member initialization of class Person
113 int Person::employee_ages[100];
114 int Person::employee_count = 0;
115
116 // driver code
117 int main(){
118     // an array of objects
119     Person employees[10] = {
120         Person("John", 25, "USA", 10000, 2000, 1000, 1000, 1000),
121         Person("Smith", 30, "UK", 20000, 4000, 2000, 2000, 2000),
122         Person("David", 35, "Canada", 30000, 6000, 3000, 3000, 3000),
123         Person("Rahul", 40, "India", 40000, 8000, 4000, 4000, 4000),
124         Person("Raj", 45, "Pakistan", 50000, 10000, 5000, 5000, 5000),
125         Person("Rohan", 50, "Bangladesh", 60000, 12000, 6000, 6000, 6000),
126         Person("Ramesh", 55, "Nepal", 70000, 14000, 7000, 7000, 7000),
127         Person("Rajesh", 60, "Sri Lanka", 80000, 16000, 8000, 8000, 8000),
128         Person("Rakesh", 65, "Maldives", 90000, 18000, 9000, 9000, 9000),
129         Person("Ramesh", 70, "Bhutan", 100000, 20000, 10000, 10000, 10000)
130     };
131
132     employees[0].youngest_eldest_employee_age();
133 }

```

Output:

📁 /mnt/d/Ro/Do/A/S/3/S/0/Lab on 🐱 master

```

-----
Name           : John
Age            : 25
Address        : USA
Basic Salary   : 10000
House Rent Allowance : 2000
Medical Allowance : 1000
Dearness Allowance : 1000
Income Tax     : 1000
Gross Salary   : 13000

```

Name : Smith
Age : 30
Address : UK
Basic Salary : 20000
House Rent Allowance : 4000
Medical Allowance : 2000
Dearness Allowance : 2000
Income Tax : 2000
Gross Salary : 26000

Name : David
Age : 35
Address : Canada
Basic Salary : 30000
House Rent Allowance : 6000
Medical Allowance : 3000
Dearness Allowance : 3000
Income Tax : 3000
Gross Salary : 39000

Name : Rahul
Age : 40
Address : India
Basic Salary : 40000
House Rent Allowance : 8000
Medical Allowance : 4000
Dearness Allowance : 4000
Income Tax : 4000
Gross Salary : 52000

Name : Raj
Age : 45
Address : Pakistan
Basic Salary : 50000
House Rent Allowance : 10000
Medical Allowance : 5000
Dearness Allowance : 5000
Income Tax : 5000
Gross Salary : 65000

Name : Rohan
Age : 50
Address : Bangladesh
Basic Salary : 60000
House Rent Allowance : 12000
Medical Allowance : 6000
Dearness Allowance : 6000
Income Tax : 6000
Gross Salary : 78000

Name : Ramesh
Age : 55
Address : Nepal
Basic Salary : 70000
House Rent Allowance : 14000
Medical Allowance : 7000
Dearness Allowance : 7000
Income Tax : 7000
Gross Salary : 91000

Name : Rajesh
Age : 60
Address : Sri Lanka
Basic Salary : 80000
House Rent Allowance : 16000
Medical Allowance : 8000
Dearness Allowance : 8000
Income Tax : 8000
Gross Salary : 104000

Name : Rakesh
Age : 65
Address : Maldives
Basic Salary : 90000
House Rent Allowance : 18000
Medical Allowance : 9000
Dearness Allowance : 9000
Income Tax : 9000
Gross Salary : 117000

Name : Ramesh
Age : 70
Address : Bhutan
Basic Salary : 100000
House Rent Allowance : 20000
Medical Allowance : 10000
Dearness Allowance : 10000
Income Tax : 10000
Gross Salary : 130000

Youngest employee age: 25

Eldest employee age: 70

3. Aim: Write a program to find the area (function name AREA) of circle, rectangle and triangle by **Function overloading** concept.


Software Used:




- **IDE:** Visual Studio Code
- **Terminal:** Windows Terminal
- **Shell:** zsh
- **Compiler:** gcc

Program:

```
1  #include<iostream>
2  using namespace std;
3
4  // defining overloaded functions to calculate area of circle, rectangle and triangle
5  int area(float radius);
6  int area(float length, float breadth);
7  int area(double base, double height);
8
9
10 // driver
11 int main()
12 {
13     int circleArea, rectangleArea, triangleArea;
14     circleArea = area(56);
15     rectangleArea = area(15.0f, 20.0f);
16     triangleArea = area(15.0, 20.0);
17
18     cout << "Area of circle: " << circleArea << endl;
19     cout << "Area of Rectangle: " << rectangleArea << endl;
20     cout << "Area of Triangle: " << triangleArea << endl;
21
22     return 0;
23 }
24
25 int area(float radius){
26     // this calculates the area of a circle with given radius
27     return 3.14 * radius*radius;
28 }
29
30 int area(float length, float breadth){
31     // this calculates the area of a rectangle with given length and breadth
32     return length * breadth;
33 }
34
35 int area(double base, double height){
36     // this calculates area of the triangle with given base and height
37     return 0.5 * base * height;
38 }
```

Output:

A terminal window header bar with a light blue background. It contains a circular icon on the left, a file path in the middle, and a branch name with a warning icon on the right.

 /mnt/d/Ro/Do/A/S/3/S/0/Lab on  master 

Area of circle: 9847
Area of Rectangle: 300
Area of Triangle: 150

4.1 Aim: Write a program to swap two numbers (create two classes) by using **Friend function**.

Software Used:

- **IDE:** Visual Studio Code
- **Terminal:** Windows Terminal
- **Shell:** zsh
- **Compiler:** gcc

Program:

```
1  #include<iostream>
2  using namespace std;
3
4  class Computer1;
5  class Computer2;
6
7  void swap(Computer1 &c1, Computer2 &c2);
8
9  class Computer1{
10     private:
11         int ram;
12
13     public:
14         Computer1(int ram){
15             this->ram = ram;
16         }
17
18         friend void swap(Computer1* a, Computer2* b);
19
20         int spec(){
21             return ram;
22         }
23 };
24
25 class Computer2{
26     private:
27         int ram;
28
29     public:
30         Computer2(int ram){
31             this->ram = ram;
32         }
33
34         friend void swap(Computer1* a, Computer2* b);
35
36         int spec(){
37             return ram;
38         }
39 };
40
```

```

40
41 void swap(Computer1* a, Computer2* b){
42     int temp = a->ram;
43     a->ram = b->ram;
44     b->ram = temp;
45 }
46
47
48 //driver code
49 int main(){
50     Computer1 c1(8);
51     Computer2 c2(16);
52
53     cout << "Before swap: " << endl;
54     cout << "c1: " << c1.spec() << endl;
55     cout << "c2: " << c2.spec() << endl << endl;
56
57     swap(&c1, &c2);
58
59     cout << "After swap: " << endl;
60     cout << "c1: " << c1.spec() << endl;
61     cout << "c2: " << c2.spec() << endl;
62
63     return 0;
64 }

```

Output:



Before swap:

c1: 8

c2: 16

After swap:

c1: 16

c2: 8

4.2 Aim: Write a program to create two classes DistM and DistF which store the values of distance. DistM stores distance in meters and centimetres and DistF stores distances in feet and inches. Read values for the class object and add one object of DistM with another object of DistF. Use a **friend function** for the addition operation and display answer in meter and centimetres.

Software Used:

- **IDE:** Visual Studio Code
- **Terminal:** Windows Terminal
- **Shell:** zsh
- **Compiler:** gcc

Program:

```
1  #include<iostream>
2  using namespace std;
3
4  class DistM;
5  class DistF;
6
7  void add(DistM* m, DistF* f);
8
9
10 // Distance in meters and centimeters
11 class DistM{
12     private:
13         int meter;
14         int centimeter;
15
16     public:
17         DistM(int meter, int centimeter){
18             this->meter = meter;
19             this->centimeter = centimeter;
20         }
21
22         friend void add(DistM* m, DistF* f);
23
24         void display(){
25             cout << "Distance: " << meter << "m " << centimeter << "cm" << endl;
26         }
27 };
28
29 // Distance in feet and inches
30 class DistF{
31     private:
32         int feet;
33         int inch;
34
35     public:
36         DistF(int feet, int inch){
37             this->feet = feet;
38             this->inch = inch;
39         }
39 }
```

```

40
41     friend void add(DistM* m, DistF* f);
42
43     void display(){
44         cout << "Distance: " << feet << "ft " << inch << "in" << endl;
45     }
46 };
47
48
49 // add two distances and display in meters and centimeters
50 void add(DistM* m, DistF* f){
51     int totalCentimeters = (m->meter * 100 + m->centimeter) + (f->feet * 30.48 + f->inch * 2.54);
52     int totalMeters = totalCentimeters / 100;
53     totalCentimeters = totalCentimeters % 100;
54
55     cout << "Total distance: " << totalMeters << "m " << totalCentimeters << "cm" << endl;
56 }
57
58
59 // driver code
60 int main(){
61     DistM m(1, 50);
62     DistF f(5, 6);
63
64     cout << "Distance 1: " << endl;
65     m.display();
66     cout << "Distance 2: " << endl;
67     f.display();
68
69     cout << endl;
70     add(&m, &f);
71
72     return 0;
73 }
74

```

Output:

```

🔄 /mnt/d/Ro/Do/A/S/3/S/0/Lab on 🐱 master ! ?
Distance 1:
Distance: 1m 50cm
Distance 2:
Distance: 5ft 6in

Total distance: 3m 17cm

```