**OOP Practicals**

1. **Merging 2 arrays, Sorting an array, Searching a number from an array**

**1.Code -**

#include<iostream>

int main()

{

int arr1[5], arr2[5], arr3[10];

int i, k, pos;

std::cout << "Enter elements for First Array - ";

for(i=0; i<5; i++)

{

std::cin >> arr1[i];

arr3[i] = arr1[i];

}

k = i;

std::cout << "Enter elements for Second Array - ";

for(i=0; i<5; i++)

{

std::cin >> arr2[i];

arr3[k] = arr2[i];

k++;

}

std::cout << "\nMerged Array is -\n";

for(i=0; i<k; i++)

{

std::cout << arr3[i] << ", ";

}

std::cout << "\nEnter position (from 0) of element you want to search = ";

std::cin >> pos;

for(i=0; i<k; i++)

{

if(i == pos)

{

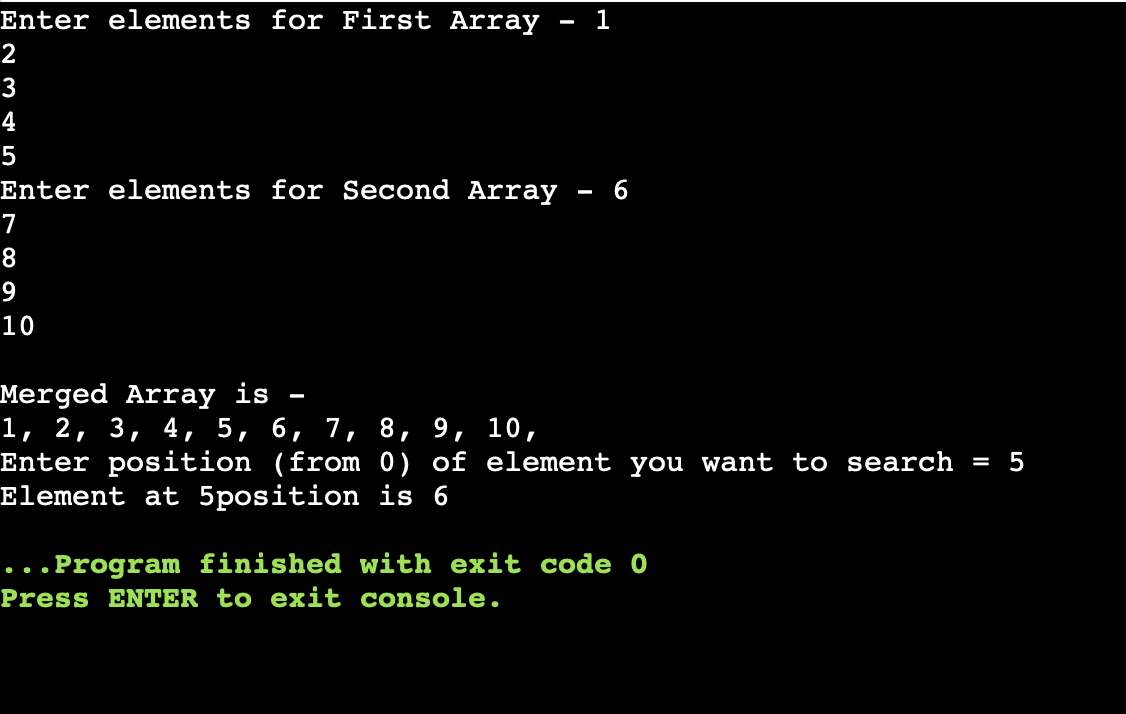
std::cout << "Element at " << i << "position is " << arr3[i];

}

}

}

**1.Output –**



1. **Call by value and Call by reference**

**2.Code –**

#include <iostream>

void change(int data) //call by value

{

data = 5;

}

void swap(int \*x, int \*y) //call by reference

{

int swap;

swap=\*x;

\*x=\*y;

\*y=swap;

}

int main()

{

int data = 3;

change(data);

std::cout << "Value of the data is: " << data;

int x=50, y=10;

std::cout << "\nValue of x before swapping: " << x;

std::cout << "\nValue of y before swapping: " << y;

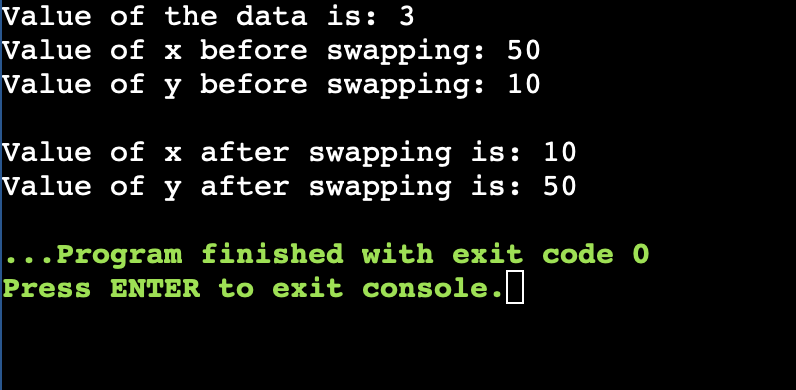
swap(&x, &y); // passing value to function

std::cout << "\n\nValue of x after swapping is: "<< x;

std::cout << "\nValue of y after swapping is: "<< y;

}

**2.Output –**



1. **Global and local variables**

**3.Code –**

#include <iostream>

// Global variable declaration

float g = 3.14;

int main()

{

// Local variable declaration

int a = 10, b = 20;

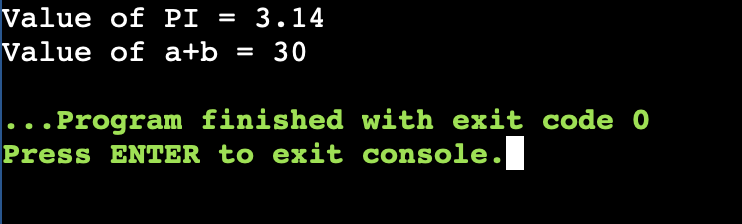
std::cout << "Value of PI = " << g; //can print here

g=a+b;

std::cout << "\nValue of a+b = " << g; //as well as change it's value anywhere

}

**3.Output –**



1. **Finding average marks of a class**

**4.Code –**

#include <iostream>

int main()

{

int n,i;

float sum=0,avg;

float\* N = new float;

std::cout << "Enter the number of Students: ";

std::cin >> n;

for (i=0;i<n;++i)

{

std::cout<<"Marks for Student " << i+1 << " = ";

std::cin >> N[i];

sum = sum + N[i];

}

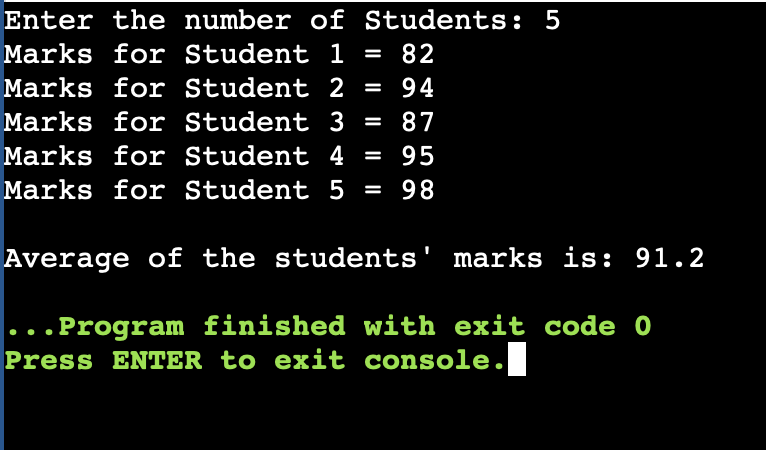
avg = sum/n;

std::cout << "\nAverage of the students' marks is: " << avg;

delete N;

}

**4.Output –**



1. **Inline function for finding volume**

**5.Code –**

#include <iostream>

#include <math.h>

inline float Vcube()

{

float a;

std::cout << "Enter edge of cube = ";

std::cin >> a;

return pow(a,3);

}

inline float Vsphere(){

float b;

std::cout << "\nEnter radius of sphere = ";

std::cin >> b;

return 4\*3.14\*pow(b,3)/3;

}

inline float Vcylin()

{

float r,h;

std::cout << "\nEnter radius of cylinder = ";

std::cin >> r;

std::cout << "\nEnter height of cylinder = ";

std::cin >> h;

return 3.14\*h\*pow(r,2);

}

inline float Vcone()

{

float r,h;

std::cout << "\nEnter radius of cone = ";

std::cin >> r;

std::cout << "\nEnter height of cone = ";

std::cin >> h;

return 3.14\*h\*pow(r,2)/3;

}

int main()

{

float finalcube = Vcube();

std::cout << "Volume of cube is = " << finalcube;

float finalS = Vsphere();

std::cout << "Volume of sphere is = " << finalS;

float finalV = Vcylin();

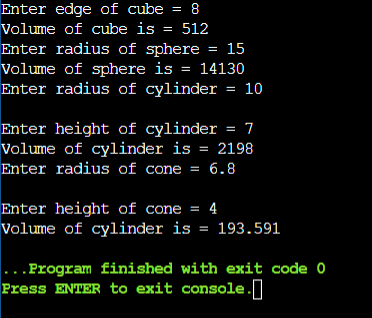
std::cout << "Volume of cylinder is = " << finalV;

float finalcone = Vcone();

std::cout << "Volume of cylinder is = " << finalcone;

}

**5.Output –**



1. **Electric bill calculation using class**

**6.Code –**

#include <iostream>

#include <string.h>

class Bill

{

char cname[50];

long int cnumber{};

char ctype[15];

long int currentmr{};

long int lastmr{};

char month[10];

long int amount{};

public:

void readData()

{

std::cout << "Write consumer name = ";

std::cin >> cname;

std::cout << "Write consumer number = ";

std::cin >> cnumber;

std::cout << "Write consumer type (out of Commercial or non-C) = ";

std::cin >> ctype;

std::cout << "Write Current meter reading = ";

std::cin >> currentmr;

std::cout << "Write Last meter reading = ";

std::cin >> lastmr;

std::cout << "Write Bill Month = ";

std::cin >> month;

}

int calculateBill()

{

long int meter{currentmr - lastmr};

if(strcmp(ctype,"Commercial"))

{

if (meter<=200)

{

amount = 5\*meter;

}

else

{

amount = 10\*meter;

}

}

if(strcmp(ctype,"Non-commercial"))

{

if (meter<=100)

{

amount = 3\*meter;

}

else

{

amount = 5\*meter;

}

}

return 0;

}

void printBill(){

std::cout << "\n Printing Bill......";

std::cout << "\nComsumer Name: " << cname;

std::cout << "\nConsumer Number: " << cnumber;

std::cout << "\nConsumer Type: " << ctype;

std::cout << "\nCurrent Meter Reading: " << currentmr;

std::cout << "\nLast Meter Reading: " << lastmr;

std::cout << "\nBill Month: " << month;

std::cout << "\nTotal Amount: " << amount;

std::cout << "\nPlease pay Bill at the earliest!!";

}

};

int main()

{

std::cout<<"Hello World\n";

Bill b;

b.readData();

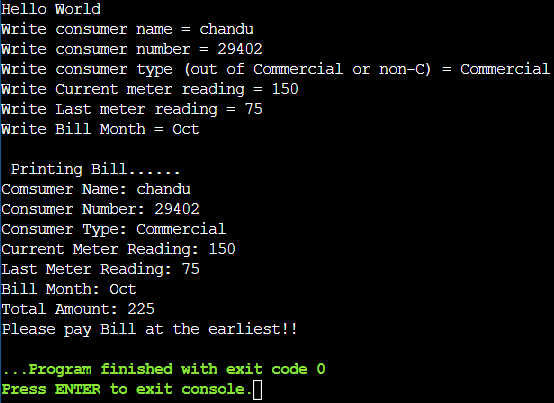
b.calculateBill();

b.printBill();

return 0;

}

**6.Output –**



1. **Calculating cricket score**

**7.Code –**

#include <iostream>

#include <string.h>

class Cricket

{

char pname[40];

char region[20];

float battingavg;

float bowlingavg;

public:

void readData()

{

std::cout << "\n Enter a Player Name: ";

std::cin >> pname;

std::cout << "\n Enter the Name of their region= ";

std::cin >> region;

std::cout << "\n Enter their Batting Average= ";

std::cin >> battingavg;

std::cout << "\n Enter their Bowling Average= ";

std::cin >> bowlingavg;

}

void generateList(Cricket player[], int i)

{

char commonplayer[i][40];

int k=0;

for(int j=0;j<i;j+=1)

{

if(player[j].battingavg > 30 && player[j].bowlingavg < 25)

{

strcpy(commonplayer[k],player[j].pname);

k+=1;

}

}

std::cout << "List of Common Players are- ";

for(int j=0;j<k;j+=1)

{

std::cout << "\n" << j+1 << ". " << commonplayer[j];

}

}

void sortList(Cricket player[], int i)

{

int op;

std::cout << "\nSort based on 1. Batting Avg. or 2. Bowling Avg.? ";

std::cin >> op;

Cricket ctemp;

if(op == 1)

{

//sort based on batting Average

for(int j=0;j<i;j+=1)

{

for(int k=j+1; k<=i; k+=1)

{

if(player[j].battingavg < player[k].battingavg)

{

ctemp = player[j];

player[j] = player[k];

player[k] = ctemp;

}

}

}

}

//sort based on bowling Average

if(op == 2)

{

for(int j=0;j<i;j+=1)

{

for(int k=j+1; k<=i; k+=1)

{

if(player[j].bowlingavg < player[k].bowlingavg)

{

ctemp = player[j];

player[j] = player[k];

player[k] = ctemp;

}

}

}

}

std::cout << "\nSorted list!! Use Display to view it.";

}

void displayList(Cricket player[], int i)

{

std::cout << "Data:- ";

for(int j=0;j<i;j+=1)

{

std::cout << "\nName: " << player[j].pname;

std::cout << "\nRegion: " << player[j].region;

std::cout << "\nBatting Average: " << player[j].battingavg;

std::cout << "\nBowling Average: " << player[j].bowlingavg;

}

}

};

int main()

{

int max;

std::cout << "Enter number of players = ";

std::cin >> max;

Cricket player[max], ptemp;

int op,i=0;

while(1)

{

std::cout << "Enter option-\n 1. Enter data of new player\n 2. Display Data of players\n 3. Exit\n 4. Sort List\n 5. Generate List of Common Players\n Enter option = ";

std::cin >> op;

if(op == 1)

{

if(i>=max)

{

std::cout << "List is Full!";

}

else

{

player[i].readData();

i+=1;

}

}

if(op == 2)

{

ptemp.displayList(player, i);

}

if(op == 3)

{

std::cout << "Breaking out.....";

break;

}

if(op == 4)

{

ptemp.sortList(player, i);

}

if(op == 5)

{

ptemp.generateList(player, i);

}

else

{

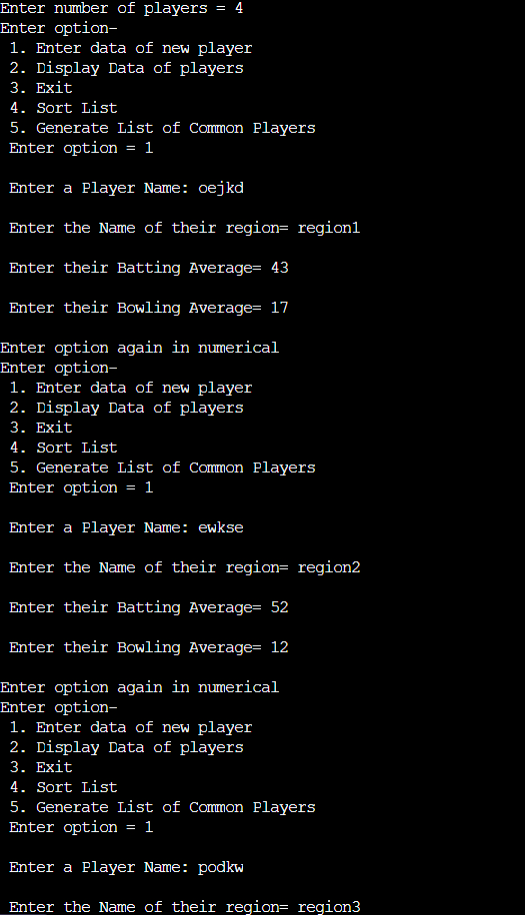
std::cout << "\nEnter option again in numerical\n";

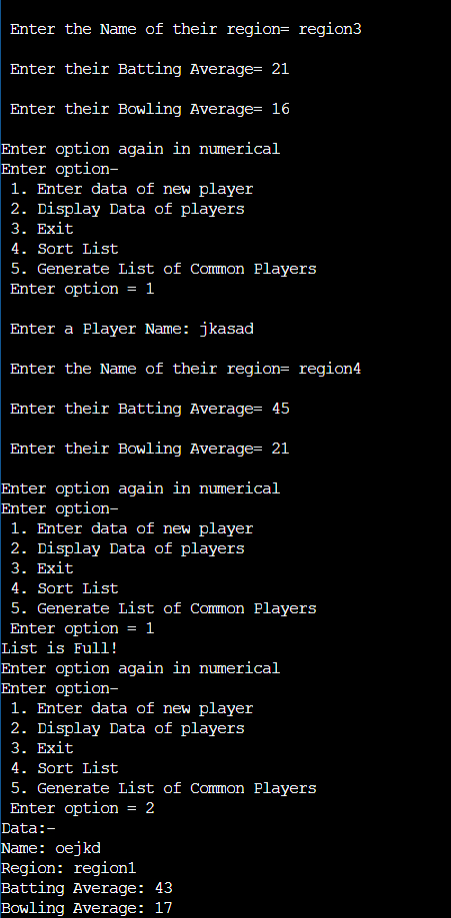
}

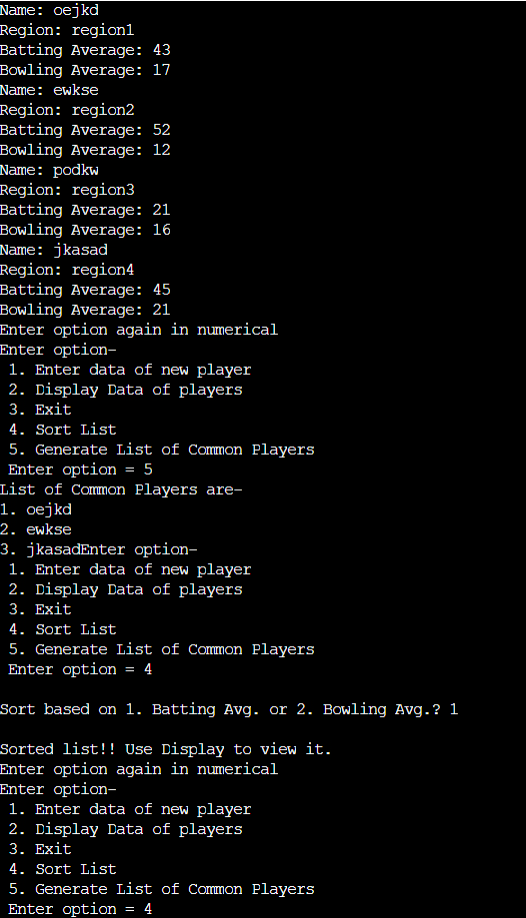
}

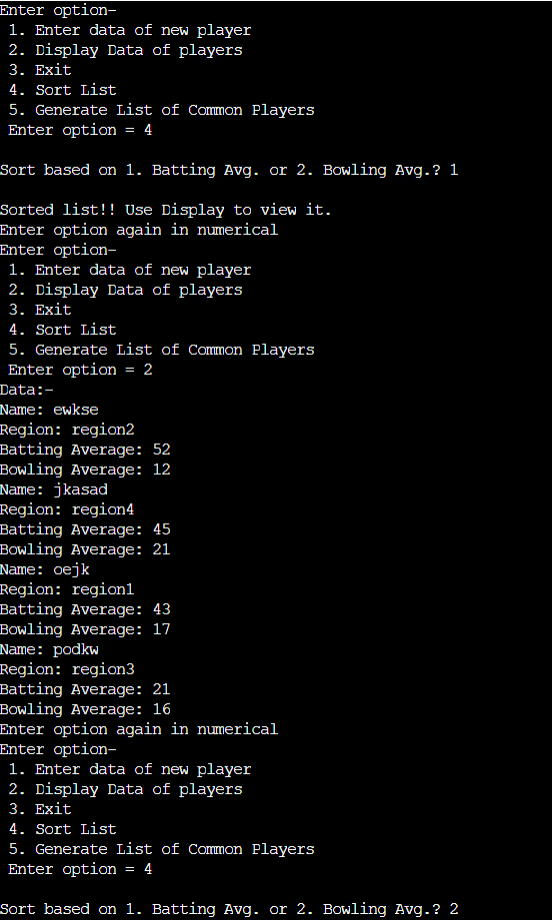
}

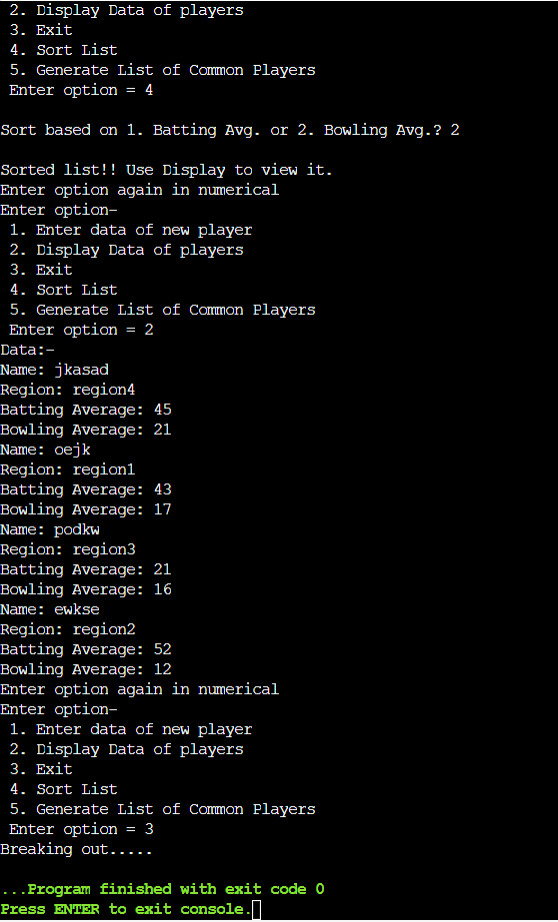
**7.Output –**











1. **Calculating ICA, TEE marks using friend function**

8**.Code** –

#include <iostream>

class TEE;

class ICA

{

float mathmarks;

float oopmarks;

float semarks;

float dsmarks;

public:

void inputmarks()

{

std::cout << "ICA marks for Maths are = ";

std::cin >> mathmarks;

std::cout << "ICA marks for OOP are = ";

std::cin >> oopmarks;

std::cout << "ICA marks for SE are = ";

std::cin >> semarks;

std::cout << "ICA marks for DS are = ";

std::cin >> dsmarks;

}

friend int calculateresult(ICA, TEE);

};

class TEE

{

float mathmarks;

float oopmarks;

float semarks;

float dsmarks;

public:

void inputmarks()

{

std::cout << "\nTEE marks for Maths are = ";

std::cin >> mathmarks;

std::cout << "TEE marks for OOP are = ";

std::cin >> oopmarks;

std::cout << "TEE marks for SE are = ";

std::cin >> semarks;

std::cout << "TEE marks for DS are = ";

std::cin >> dsmarks;

}

friend int calculateresult(ICA, TEE);

};

int calculateresult(ICA a, TEE b)

{

std::cout << "\nTotal Marks out of 100 subject wise-";

std::cout << "\nMaths: " << a.mathmarks+b.mathmarks;

std::cout << "\nOOP: " << a.oopmarks+b.oopmarks;

std::cout << "\nSE: " << a.semarks+b.semarks;

std::cout << "\nSE: " << a.dsmarks+b.dsmarks;

float finalresult = (a.mathmarks+b.mathmarks+a.oopmarks+b.oopmarks+a.semarks+b.semarks+a.dsmarks+b.dsmarks)/5;

std::cout << "\nFinal Result: " << finalresult << "%";

return 0;

}

int main()

{

std::cout << "Hello World\n";

ICA a;

TEE b;

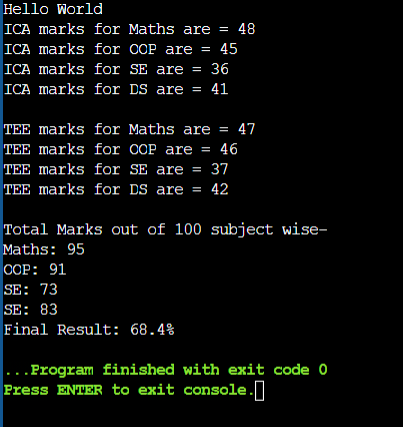
a.inputmarks();

b.inputmarks();

calculateresult(a, b);

}

**8.Output –**



1. **Evaluating Employee sales**

**9.Code –**

#include <iostream>

class Employee

{

private:

char employeename[40];

int employeeid;

char edepartment[20];

float salary;

public:

void readData()

{

std::cout << "Enter Employee name = ";

std::cin >> employeename;

std::cout << "Enter Employee ID number = ";

std::cin >> employeeid;

std::cout << "Enter Employee department name = ";

std::cin >> edepartment;

std::cout << "Enter Employee salary in USD= ";

std::cin >> salary;

}

void display()

{

std::cout << "\nThe Employee details are -";

std::cout << "\nName: " << employeename;

std::cout << "\nID: " << employeeid;

std::cout << "\nDepartment: " << edepartment;

std::cout << "\nSalary: " << salary;

}

friend class Sales;

};

class Sales

{

float sales;

public:

void readsales()

{

std::cout << "\nEnter the employee's sales this month = ";

std::cin >> sales;

}

void incentives(Employee t)

{

if(sales>=50 && sales<=100)

{

std::cout << "\nIncentives based on sales this month are = " << t.salary/10;

}

if(sales>=100 && sales<=150)

{

std::cout << "\nIncentives based on sales this month are = " << t.salary/5;

}

if(sales>150)

{

std::cout << "\nIncentives based on sales this month are = " << 3\*t.salary/10;

}

}

void performance()

{

if(sales>=50 && sales<=100)

{

std::cout << "\nSales this month were 'Satisfactory!!', Needs more improvement!";

}

if(sales<50)

{

std::cout << "\nSales this month were 'Poor!!', Work Harder! ";

}

if(sales>=101 && sales<=150)

{

std::cout << "\nSales this month were 'Good!!', Keep it up!";

}

if(sales>150)

{

std::cout << "\nSales this month were 'Excellent!!', Too bad you only get 30 percent pay raise :( ";

}

}

};

int main()

{

std::cout << "Hello World\n";

Employee a;

Sales b;

a.readData();

a.display();

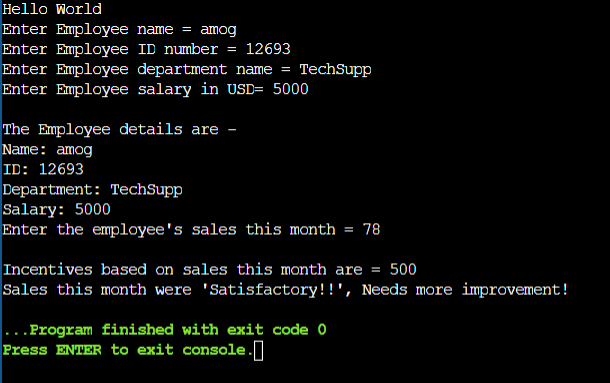
b.readsales();

b.incentives(a);

b.performance();

}

**9.Output –**



1. **Employee salary using constructors**

**10.Code –**

#include <iostream>

class Employee

{

char ename[40];

int id;

char department[40];

float salary;

float DA;

float HRA;

float TA;

float GrossSalary;

public:

Employee(){ //constructor

std::cout << "Enter Employee ID to proceed = ";

}

Employee(int a) //constructor overloading

{

id = a;

std::cout << "\nEnter further Employee Details-";

std::cout << "\nEnter Employee Name = ";

std::cin >> ename;

std::cout << "\nEnter Employee Department Name = ";

std::cin >> department;

std::cout << "\nEnter Employee Basic Salary = ";

std::cin >> salary;

}

void display()

{

DA = salary/2;

HRA = 0.3\*salary;

TA = salary/10;

GrossSalary = salary + DA + HRA + TA;

std::cout << "Employee Details - ";

std::cout << "\nName: " << ename;

std::cout << "\nID: " << id;

std::cout << "\nDepartment: " << department;

std::cout << "\nBasic Salary: " << salary;

std::cout << "\nDearness Allowance: " << DA;

std::cout << "\nHRA: " << HRA;

std::cout << "\nTA: " << TA;

std::cout << "\nGross Salary: " << GrossSalary;

}

~Employee()

{

std::cout << "\nDestructor called here, function ended.";

} // two objects so execute twice

};

int main()

{

int x;

std::cout << "Hello World\n";

Employee obj1;

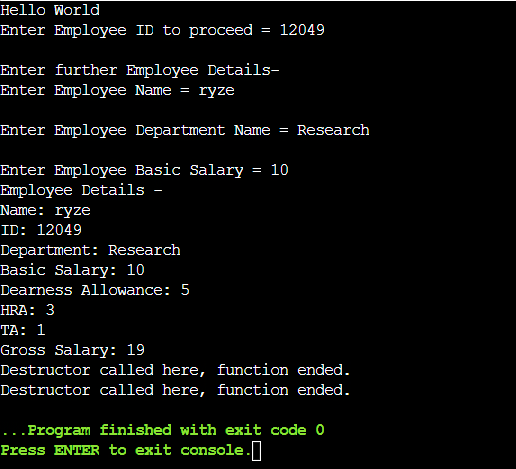
std::cin >> x;

Employee obj2(x);

obj2.display();

}

**10.Output –**



1. **Friend and operator overloading**

**11.Code –**

#include <iostream>

class Doctor

{

char doctor[20];

char area[20];

public:

void dinput()

{

std::cout << "\nEnter Name = ";

std::cin >> doctor;

std::cout << "\nEnter Area = ";

std::cin >> area;

}

Doctor operator < (Doctor &x)

{

Doctor y;

if(x.area < y.area)

{

return TRUE;

}

else

{

return FALSE;

}

}

};

int main()

{

int size, i, j;

std::cout << "Enter size of list of doctors = ";

std::cin >> size;

Doctor a[size];

std::cout << "\nNow enter details of MCGM Hospital - ";

for(i=0; i < size; i+=1)

{

std::cout << "\nEnter details for doctor " << i+1 << " - ";

a[i].dinput();

}

for(i = 1; i < size; i+=1)

{

j = i - 1;

while(a[i] < a[j] && j>=0)

{

a[j + 1] = a[j];

j = j - 1;

}

a[j + 1]=a[i];

}

}

1. **New and Delete**

**12.Code –**

#include <iostream>

void display(char name[], int rollno, int dob, int marks, char divi[] = "B2", char prog[] = "CSBS")

{

std::cout << "\n\n Name: " << name << "\n Roll No.: " << rollno << "\n Date of Birth: " << dob << "\n Marks: " << marks << "\n Division: " << divi << "\n Program: " << prog;

}

inline void sinput()

{

char name[20];

std::cout << "\nEnqter Name of the student = ";

std::cin >> name;

int \*rollno = new int;

std::cout << "Enter Roll no. of the student = ";

std::cin >> \*rollno;

int \*dob = new int;

std::cout << "Enter Date of birth of the student = ";

std::cin >> \*dob;

int \*marks = new int;

std::cout << "Enter Marks of the student = ";

std::cin >> \*marks;

display(name,\*rollno,\*dob,\*marks);

delete rollno;

delete dob;

delete marks;

}

int main()

{

std::cout << "Enter details for student 1-";

sinput();

std::cout << "\nEnter details for student 2-";

sinput();

std::cout << "\nEnter details for student 3-";

sinput();

}

12.Output – 