**Task 2:**

#include<iostream>

#include<string>

using namespace std;

class Person

{

private:

string name;

const string date;

int cnic;

string status;

int month;

int day;

int year;

public:

int getcnic( int cnic);

const void getdob();

const void display();

void getname(string name);

void output(int month, int day, int year);

void setdate(int m, int d, int y);

};

void Person::setdate(int m, int d, int y)

{

month = m;

month = (m >= 1 && m <= 12) ? m : 0;

day = (d >= 1 && d < 31) ? d : 0;

year = (y >= 0) ? y : 0;

getname(name);

}

void Person::output(int month, int day, int year)

{

switch (month)

{

case 1:

{

cout << day << " january " << " " << year;

break;

}

case 2:

{

cout << day << " february " << " " << year;

break;

}

case 3:

{

cout << day << " march " << " " << year;

break;

}

case 4:

{

cout << day << " april " << " " << year;

break;

}

case 5:

{

cout << day << " may " << " " << year;

break;

}

case 6:

{

cout << day << " june " << " " << year;

break;

}

case 7:

{

cout << day << " july " << " " << year;

break;

}

case 8:

{

cout << day << " august " << " " << year;

break;

}

case 9:

{

cout << day << " september " << " " << year;

break;

}

case 10:

{

cout << day << " october " << " " << year;

break;

}

case 11:

{

cout << day << " november " << " " << year;

break;

}

case 12:

{

cout << day << " december " << " " << year;

break;

}

default:

cout << "wrong getdob ";

break;

}

}

const void Person::getdob()

{

int m, d, y;

cout << "enter months : ";

cin >> m;

cout << "\nenter day : ";

cin >> d;

cout << "\nenter year : ";

cin >> y;

month = m;

day = d;

year = y;

setdate(m, d, y);

}

int Person:: getcnic( int cnic)

{

cout << "\n\n\nenter cnic";

cin >> cnic;

cout << cnic;

return cnic;

}

void Person::getname(string name)

{

cout << "enter name";

cin >> name;

getcnic(cnic);

cout << "\n Mr." << name<<" ";

output(month, day, year);

cout << " "<< cnic;

}

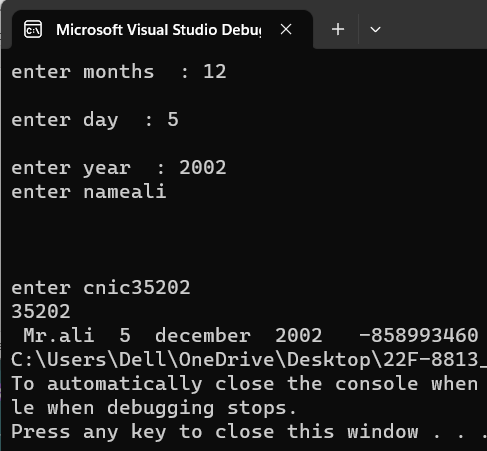
int main()

{

Person ob;

ob.getdob();

}



**Task 1:**

#include <iostream>

#include <string>

using namespace std;

class employ {

public:

employ() : bankaccount("123456") {}

employ(string emid, string emname, string emdp, string emgrade) : id(emid), name(emname),

department(emdp), bankaccount("123456"), Grade(emgrade) {}

void displayData() const

{

cout << "id is : " << id << endl;

cout << "name is: " << name << endl;

cout << "department is: " << department << endl;

cout << "Bank Account Number is: " << bankaccount << endl;

cout << "Grade is: " << Grade << endl;

}

void inputData()

{

cout << "Enter the id : ";

getline(cin, id);

cout << "Enter the name: ";

getline(cin, name);

cout << "Enter the department: ";

getline(cin, department);

cout << "Enter the Grade : ";

getline(cin, Grade);

}

~employ()

{

cout << "Destructor is called for the employ number: " << name << endl;

}

private:

string department;

const string bankaccount;

string id;

string name;

string Grade;

};

int main()

{

employ\* ptr1 = new employ();

employ\* ptr2 = new employ();

employ\* ptr3 = new employ();

ptr1->inputData();

ptr2->inputData();

ptr3->inputData();

ptr1->displayData();

ptr2->displayData();

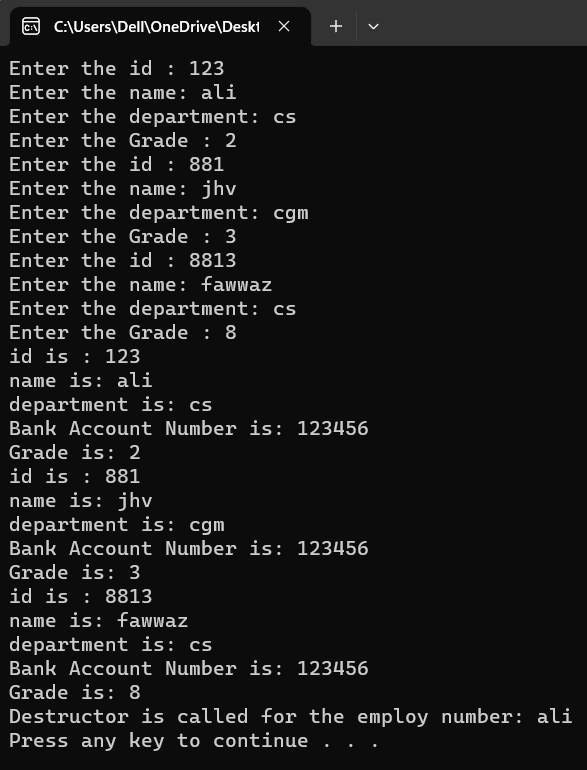
ptr3->displayData();

delete ptr1, ptr2,ptr3;

system("pause");

return 0;

}



**Task 3:**

//#include<iostream>

//using namespace std;

//int main()

//{

////a

// // int \*number;

// //cout << number << endl;

// // we haven't initialised as well as we have not given the address to the pointer to which it points

// //correction

// //int\* number,num=0;//if we initializes pointer with 0 it gives 0000000's as address

// //number = &num;

// //cout << number << endl;

////-----------------------------------------------------------------------------------------

//

////b)

////double\* realPtr=0;

//// // long \*integerPtr;

//// //correction

//// double\* integerPtr;

//// integerPtr = realPtr;//this is wrong datatype must be same and we have to initialize it

//// //or assign a address of some num or other

//

//

////---------------------------------------------------------------------------------

// // c)

// //int \* x, y=0;

// //x = &y;

// ////we have to assign address and initialize it as well

//

//

////---------------------------------------------------

//

//// // d)

//// char s[] = "this is a character array";

//// int i = 0;

////for (;\*s != '\0'; ++i)

////cout << \*s << ' ';

//

////----------------------------------------------------------------------------

////e)

//// short \*numPtr,result;

//// void \*genericPtr = numPtr;

//// short\* generiPtr=0;

////result = \*generiPtr + 7;

//////error is that void pointer point any dataype and when we try to equal to other datstype than it is of no datatype so any datatype could not point it

////

////------------------------------------------------------

//

//

//

//

//

//

//

//

////f)

//// double x = 19.34;

////double \*xPtr = &x;//we have to write sterrick only bcz variable doesn't holds the address

////cout << xPtr << endl;

//

//}

**Task 4:**

// //Part a)

//#include <iostream>

// using namespace std;

// void mystery1(char\*, const char\*); // prototype

// int main()

// {

// char string1[80];

// char string2[80];

// cout << "Enter two strings: ";

// cin >> string1 >> string2;

// mystery1(string1, string2);// we called a func in which string 1 is copied or attached with constant stirng 2

// cout << string1 << endl;

// } // end main

// // What does this function do?

// void mystery1(char\* s1, const char\* s2)//char ptr s1 points to string1 and we have made the string 2 constant by using const pointer

// {

// while (\*s1 != '\0')//while loop ends at last of string

// ++s1;//it increments the indexes of array and go to last index where is null character

// for (; \*s1 = \*s2; ++s1, ++s2);//it copies string 2 into 1 starting from the index last of string 1 and 0 of string 2

// // empty statement

// } // end function mystery1

//-------------------------------------------------------------------------------------------------------------------------------

//

////Part b)

//#include <iostream>

//using namespace std;

//int mystery2(const char\*); // prototype

//int main()

//{

// char string1[80];

// cout << "Enter a string: ";

// cin >> string1;

// cout << mystery2(string1) << endl;

//} // end main

//// What does this function do?

//int mystery2(const char\* s)

//{

// int x;

// for (x = 0; \*s != '\0'; ++s)//we have initialised x with 0 which increments w.r.t to the incrementation of array indexes

// ++x;

// return x;//return size of string without space

//} // end function mystery2

**Task 5:**

#include<iostream>

#include<conio.h>

using namespace std;

int main()

{

int a = 5, b = 10;

int c;

int\* p1, \* p2;

p1 = &a;//address of a is stored in pointer1

p2 = &b;//address of b is stored in pointer2

c = \*p1;//pointer 1 value (5) is stored in c

cout << "\*(p1++) =" << \*(p1++) << endl;//it post increments the value and print value before incrementing

cout << "value of p1 " << p1 << endl;//print address after post incrementing like address of 6

cout << "\*(++p1) =" << \*(++p1) << endl;//pre increments the address due to which garbage value is sotred in the pointer value of 7

cout << "value of p1 " << p1 << endl;//print address which is of garbage after pre incrementing

cout << "(\*p1)++ =" << (\*p1)++ << endl;//it post incremnets in the value but prints the address of previous value

cout << "value of p1 " << p1 << endl;//prints the address of 5 not 6

cout << "++(\*p1) =" << ++(\*p1) << endl;//pre increment the value like 6+1=7 and prints the address of 7

cout << "value of p1 " << p1 << endl;//prints the address of 7

return 0;

}