

PuRu
\$!Ngh

Program :- 31

Write a cpp program in which use constructors in derived class.

```
#include <conio.h>
#include <iostream.h>
class alpha
{
    int x;
public:
    alpha(int i)
    {
        x = i;
        cout << "alpha initialized:" << endl;
    }
    void showx()
    {
        cout << "x: " << x << endl;
    }
};
```

```
class beta
{
public:
    beta(float j)
    {
        y = j;
        cout << "beta initialized:" << endl;
    }
};
```

```
void showy()
{
    cout << "y: " << y << endl;
};
```

```
class Gamma: public bete, public cipher
```

```
{ int m, n;
```

```
public:
```

```
Gamma(int a, float b, int c, int d):  
    cipher(a), bete(b)
```

```
m=c;
```

```
n=d;
```

```
cout << "Gamma initialized:";
```

```
}
```

```
void showmn()
```

```
cout << "m: " << m << "\n";
```

```
cout << "n: " << n << "\n";
```

```
};
```

```
main()
```

```
Gamma g(5, 10.75, 20, 30);
```

```
g.showm();
```

~~g.showc();~~~~g.showY();~~~~g.Showmn();~~~~getch();~~

```
return 0;
```

```
}
```

Output : \downarrow

beta initialized:

alpha initialized:

gamma initialized:

$x: 5$

$y: 10.75$

$m: 20$

$n: 30$

Dinesh

22/10/13

Program :- 32

Write a C++ program
of initialization list in Constructors.

```
#include <conio.h>
#include <iostream.h>
class alpha
{
    int x;
public:
    alpha(int i)
    {
        x = i;
        cout << "alpha initialized";
    }
    void showcipher()
    {
        cout << "x:" << x << "\n";
    }
};

class beta
{
    float p, q;
public:
    beta(float a, float b): p(a), q(b+p)
    {
        cout << "beta initialized";
    }
    void showbeta()
    {
        cout << "p:" << p << "\n";
        cout << "q:" << q << "\n";
    }
};
```

```

class Gamma : public Beta, public Alpha
{
    int u, v;
public:
    Gamma(int a, int b, float c):
        cipher(a * e), beta(c, c), u(a)
    {
        cout << "Gamma initialized";
    }
    void showGamma()
    {
        cout << "u: " << u << endl;
        cout << "v: " << v << endl;
    }
};

main()
{
    Gamma g(2, 4, 2.5);
    cin >> n;
    g.showcipher();
    g.showbeta();
    g.showGamma();
    getch();
    return 0;
}

```

Output :

better initialized
cipher initialized
grammar initialized

X : 4

P : 2.5

q : 5

u : 2

v : 4

Dinesh

22/10/15

Program : 33

Write a cpp program for implementation of pointers to objects.

```
#include <conio.h>
#include <iostream.h>
class item
{
    int code;
    float price;
public:
    void getdata(int a, float b)
    {
        code = a;
        price = b;
    }
    void show()
    {
        cout << "Code: " << code << "\n";
        cout << "Price: " << price << "\n";
    }
};

const int size = 2;
main()
{
    item *P = new item[size];
    item *d = P;
    int x, i;
    float y;
    clrscr();
    for(i=0; i<size; i++)
    {
        cout << "Input code & price for item " << i+1 << "\n";
        cin >> d->code >> d->price;
        d++;
    }
}
```

```
cin >> x >> y;  
p -> getdata(x, y);  
p++;  
}  
for(i = 0; i < size; i++)  
{  
    cout << "Item is " << i+1 << ":" << endl;  
    d -> show();  
    d++;  
}  
getch();  
return 0;
```

Output :

Input code & price for item 1

1

10

Input code & price for item 2

2

20

Item is : 1

Code : 1

price : 10

Item is : 2

Code : 2

price : 20

Program :- 34

Write a CPP program for implementation of array of pointer to objects.

```
#include <iostream.h>
#include <conio.h>
#include <iostream>
class city
{
protected:
    char *name;
    int len;
public:
    city()
    {
        len = 0;
        len = new char[len + 1];
    }
    void getname()
    {
        char *s;
        s = new char[30];
        cout << "Enter city name:" ;
        cin >> s;
        len = strlen(s);
        name = new char[len + 1];
        strcpy(name, s);
    }
    void printname()
    {
        cout << name << "\n";
    }
};

main()
```

```
{  
    city *cptn[10];  
    int n = 1;  
    int option;  
    if (fscanf(stdin, "%d", &n) != 1)  
        do  
    {  
        cptn[n] = new city;  
        cptn[n] -> getname();  
        n++;  
        cout << "Do you want to enter  
        one more name? " << "\n";  
        cout << "Enter 1 for Yes 0 for no:";  
        cin >> option;  
    }  
    while (option);  
    cout << "\n\n";  
    for (int i = 1; i <= n; i++)  
    {  
        cptn[i] -> printname();  
    }  
    getch();  
    return 0;  
}
```

Output :

Enter city name: Ahmedabad

Do you want to enter 1 more name?

Enter 1 for yes 0 for no: 1

Enter city name: Baroda

Do you want to enter 1 more name?

Enter 1 for yes 0 for no: 1

Enter city name: Gandhinagar

Do you want to enter 1 more name?

Enter 1 for yes 0 for no: 0

Ahmedabad

Baroda

Gandhinagar

Aray
22/10/15

Program :- 35

Write a C++ program for implementation of this pointer

```
#include <conio.h>
#include <iostream.h>
#include <string.h>

class person
{
    char name[20];
    float age;

public:
    Person(char *s, float a)
    {
        strcpy(name, s);
        age = a;
    }

    person & person :: greater(person &x)
    {
        if (x.age >= age)
            return x;
        else
            return *this;
    }

    void display()
    {
        cout << "Name: " << name << "\n";
        cout << "Age: " << age << "\n";
    }
};
```

```
main( )
```

```
{
```

```
    person P1 ("John", 37.50);
```

```
    person P2 ("Ahmed", 29.0);
```

```
    person P3 ("Hebber", 40.25);
```

```
    cin >>
```

```
    person P = P1. greater(P3);
```

```
    cout << "Elder person is:" << "\n";
```

```
    P. display();
```

```
    P = P1. greater(P2);
```

```
    cout << "Elder person is:" << "\n";
```

```
    P. display();
```

```
    getch();
```

```
    return 0;
```

```
}
```

Output is ↓

Elder person is ;

Name : Hebber

Age : 40.25

Elder person is :

Name : John

Age : 37.5

Program :- 36

Write a cpp program for implementation of virtual function.

```
#include <conio.h>
#include <iostream.h>
class Base
{
public:
    void display()
    {
        cout << "Display Base:";
    }
    virtual void show()
    {
        cout << "\n Show base:";
    }
};

class Derived : public Base
{
public:
    void display()
    {
        cout << "\n Display derived:";
    }
    void show()
    {
        cout << "\n Show derived:";
    }
};

main()
{
    Base B;
    Derived D;
```

```
Base *bptr;  
clnsn();  
cout << "In bptr points to base class:";  
cout << endl;  
bptr = & B;  
bptr -> display();  
bptr -> Show();  
cout << endl << "bptr points to derived class:";  
cout << endl;  
bptr = & D;  
bptr -> display();  
bptr -> Show();
```

```
getch();  
return 0;
```

}

Output :-

bptr points to base class:

Display base
Show base

bptr points to derived class:

Display derived
Show derived

say

13

Program :- 37

Write a C++ program which explains a concept of runtime polymorphism

```
#include <conio.h>
#include <iostream.h>
#include <string.h>;
class media
{
protected:
    char title[50];
    float price;
public:
    media(char *s, float a)
    {
        strcpy(title, s);
        price = a;
    }
    virtual void display()
}
};

class book : public media
{
int pages;
public:
    book(char *s, float a, int p) : media(s, a)
    {
        Pages = p;
    }
    void display();
}
};
```

class Tape : public media

float Time;

public:

Tape (char *s, float a, float t) : media(s,a)

{
 time = t;
}

void display();
};

void book :: display()

cout << "In Title is :" << title;

cout << "In Pages are :" << pages;

cout << "In Price is :" << price;

void tape :: display()

cout << "In title is :" << title;

cout << "In play time is :" << time << "mins";

cout << "In price is :" << price;

main()

char *title = new char[30];

float price, time;

int pages;

cin >> title;

cout << "Enter book details :\n";

cout << "title is :";

cin >> title;

cout << "price is :";

cin >> price;

Cout << "Pages are:";
 cin >> Pages;

book b1(title, price, pages);

Cout << "Enter tape details:" << "\n";
 Cout << "Title is: " << title;
 cin >> title;
 Cout << "playtime (mins) is:";
 Cin >> Time;

Tape t1(title, price, time);

media *list[2];
 list[0] = & b1;
 list[1] = & t1;

Cout << "----- Book -----";
 list[0] -> display();

Cout << "----- Tape -----";
 list[1] -> display();

getch();

return 0;

}

Output :-

Enter book details :

Title is: c++

Price is: 300.00

Pages are: 500

Enter tape details:

Title is: ADBMS

Price is: 90

play time (mins) is: 60.00

----- Book -----

Title is: c++

Pages are: 500

Price is: 300.00

----- Tape -----

Title is: ADBMS

play time is: 60.00 mins

price is: 90

Dinesh

22/10/13