

CS - OUTPUT PROGRAMS QUESTIONS AND ANSWERS

1. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class distance
{
int feet,inches;
public:
void distance_assign(int f, int i)
{
feet=f;
inches=i;
}
void display()
{
cout<<"\n Feet:"<<feet<<"\t Inches:"<<inches;
}
distance operator+(distance d2)
{
distance d3;
d3.feet=feet+d2.feet;
d3.inches=(inches+d2.inches)%12;
d3.feet+=(inches+d2.inches)/12;
return d3;
}
};
void main()
{
clrscr();
distance dist1,dist2;
dist1.distance_assign(13,10);
dist2.distance_assign(16,4);
distance dist3=dist1+dist2;
dist1.display();
dist2.display();
dist3.display();
getch();
}
```

Output:

Feet : 13 Inches : 10
Feet : 16 Inches : 4
Feet : 30 Inches : 2

2. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class distance
{
int feet,inches;
public:
void distance_assign(int f, int i)
{
feet=f;
inches=i;
}
void display()
{
cout<<"\n Feet:"<<feet<<"\t Inches:"<<inches;
}
distance operator+(distance d2)
{
distance d3;
d3.feet=feet+d2.feet;
d3.inches=(inches+d2.inches)%12;
d3.feet+=(inches+d2.inches)/12;
return d3;
}
};
void main()
{
clrscr();
distance dist1,dist2;
dist1.distance_assign(12,11);
dist2.distance_assign(24,2);
distance dist3=dist1+dist2;
dist1.display();
dist2.display();
dist3.display();
getch();
}
```

Output:

Feet : 12 Inches : 11
Feet : 24 Inches : 2
Feet : 37 Inches : 1

3. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class distance
{
int feet,inches;
public:
void distance_assign(int f, int i)
{
feet=f;
inches=i;
}
void display()
{
cout<<"\n Feet:"<<feet<<"\t Inches:"<<inches;
}
distance operator+(distance d2)
{
distance d3;
d3.feet=feet+d2.feet;
d3.inches=(inches+d2.inches)%12;
d3.feet+=(inches+d2.inches)/12;
return d3;
}
};
void main()
{
clrscr();
distance dist1,dist2;
dist1.distance_assign(24,11);
dist2.distance_assign(12,1);
distance dist3=dist1+dist2;
dist1.display();
dist2.display();
dist3.display();
getch();
}
```

Output:

```
Feet : 24      Inches : 11
Feet : 12      Inches : 1
Feet : 37      Inches : 0
```

4. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
float area(float r)
{
cout<<"\n Circle .....";
return(22/7*r*r);
}
float area(float k, float b, float h)
{
cout<<"\n Triangle...";
return(k*b*h);
}
float area(float l, float b)
{
cout<<"\n Rectangle...";
return(l*b);
}
void main()
{
cout<<area(3.0, 4.5)
cout<<area(5.0);
cout<<area(0.5, 4.0, 6.0);
getch();
}
```

Output:

```
Rectangle .... 13.5
Circle....      75
Triangle...     12
```

5. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
void fun(char a, int times)
{
for(int i=1; i<=times; i++)
{
cout<<<a<<"\n"; }
void fun(int times=5, char a='*')
{
for(int i=1; i<=times; i++)
cout<<a<<"\n";
}
void main()
{
fun(3,'+');
fun();
}
```

Output :

```
+++
*****
```

6. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class simple
{
private:
int a,b;
public:
simple(int x, int y)
{
a=x;
b=y;
}
~simple()
{
cout<<"Destructor";
}
void putdata()
{
cout<<"A="<<a<<"\n";
cout<<"B="<<b;
}
};
void main()
{
clrscr();
simple s(5,6);
s.putdata();
getch();
}
```

Output :

A=5
B=6
Destructor

7. Write the Output for given C++ Program:

```
#include <iostream.h>
#include<conio.h>
class product
{
private:
int x,y;
public:
product()
{
x=0;
y=0;
cout<<"\n Constructor of class product...";
}
~product()
{
cout<<"\n Destructor of class product...";
}
void getdata()
{
x=5;
y=10;
}
void display()
{
cout<<"\n The two integers are ...."<<x<<"\t"<<y;
cout<<"\n The product of the integers are...."<<x*y;
}
};
```

Output :

Constructor of class product...
The two integers are 5 10
The product of the integers are.... 50
Destructor of class product...

8. Write the Output for given C++ Program:

```
# include<iostream.h>
#include<conio.h>
class add
{ int num1, num2, sum;
public:
add ( int s1, int s2 )
{
num1= s1;
num2=s2;
}
add (add &a )
{
num1= a.num1;
num2=a.num2;
}
```

```
void putdata()
{
cout<<sum;
}
void addition()
{
sum=num1+num2;
}
};
void main()
{
add b (10, 20),c(b);
b.addition();
c.addition();
b.putdata();
c.putdata();
getch();
}
```

Output : 30 30

9. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class simple
{
private:
int a,b;
public:
simple()
{
a=0;
b=0;
cout<<<"\n Welcome";
}
~simple() {
cout<<<"\n Good Bye";
}
void getdata()
{
cout<<<"\n Value of a and b are 5 and 6";
a=5; b=6;
}
void putdata()
{
cout<<<"The two integers are"<<a<<"\t"<<b;
cout<<<"\n The sum is"<<a+b;
}
};
void main()
{
simple s;
s.getdata();
s.putdata();
}
```

Output :

Welcome

Value of a and b are 5 and 6

The two integers are 5 6

The sum is 11

Good Bye

10. Write the Output for given C++ Program:

```
#include <iostream.h>
class add
{
int sum;
protected:
int n1,n2;
public:
add()
{
n1=n2=sum=0;
cout<<<"\n Add Constructor";
}
void accept()
{
n1=2;
n2=2;
}
void plus()
{
sum=n1+n2;
cout<<<"\n The sum of two nos is"<<sum;
}
};
class subtract:public add
{
int sub;
public:
subtract()
{
sub=0;
cout<<<"\n Subtract constructor..";
}
void minus()
{
add::accept();
sub=n1-n2;
cout<<<"\n The difference of two numbers is"<<sub;
}
};
void main()
{
subtract s;
s.accept();
s.plus();
s.minus();
}
```

Output:

Add constructor

Subtract constructor..

The sum of two nos is 4

The difference of two numbers is 0

11. Write the Output for given C++ Program:

```
#include <iostream.h>
# include<iostream.h>
#include<conio.h>
class add
{
int num1, num2, sum;
public:
add()
{
cout<<“\n Constructor without parameters.. “;
num1= 0;
num2= 0;
sum = 0;
}
add ( int s1, int s2 )
{
cout<<“\n Parameterized constructor... “;
num1= s1;
num2=s2;
sum=NULL;
}
void getdata()
{
cout<<“Enter data ... “;
cin>>num1>>num2;
}
void addition(add b)
{
sum=num1+num2+b.num1+b.num2;
}
void putdata() {
cout<<“\n The numbers are..”;
cout<<num1<<“\t”<<num2;
cout<<“\n The sum of the numbers are “<<“\t”<< sum;
} };
void main()
{
add a, b (10, 20);
a.getdata();
a.addition(b);
cout<<“\n Object a : “;
a.putdata();
getch();
}
```

Output :

Constructor without parameters..

Parameterized constructor...

Enter data ... 2 3

Object a :

The numbers are..2 3

The sum of the numbers are 35

12. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
class student
{
private:
char n[30];
int rno,m1,m2,total;
protected:
void assign()
{
rno=5;
strcpy(n,“Sri”);
m1=78;
m2=83;
}
void compute()
{ total=m1+m2;
}
void display()
{
cout<<“\nRollno”<<rno;
cout<<“\nName “<<n;
cout<<“\nMark1 “<<m1;
cout<<“\nMark2 “<<m2;
cout<<“\nTotal “<<total;
}
public:
void execute()
{
assign();
compute();
display();
}
};
void main()
{
student stud;
stud.execute();
}
```

Output:

Rollno 5

Name Sri

Mark1 78

Mark2 83

Total 161

13. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class add
{
int n1, n2, s;
public:
add()
{
cout<<"\n Constructor without parameters.. ";
n1=n2=s=0;
}
add ( int s1, int s2 )
{
cout<<"\n Parameterized constructor... ";
n1= s1;
n2=s2;
}
add (add &a){
cout<<"\n Copy Constructor ... ";
n1= a.n1;
n2=a.n2;
}
void assign()
{
n1=3;
n2=2;
}
void addition(add b)
{
s=b.n1 + b.n2;
}
add addition() {
add a(5,6);
sum = num1 + num2 +a.num1 +a.num2;
}
void put() {
cout<<"\n The numbers are.."<<n1<<"\t"<<n2;
cout<<"\n Sum... "<< s;
}
};
void main()
{
clrscr();
add a, b (15, 20);
a.assign();
a.addition(a);
a.addition(b);
cout<<"\n Object a : ";
a.put();
cout<<"\n Object b : ";
b.put();
getch();
}
```

Output:

Constructor without parameters..

Parameterized constructor..

Copy Constructor ...

Copy Constructor ...

Object a :

The numbers are..3 2

Sum... 5

Object b :

The numbers are..15 20

Sum... 35

14. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class simple
{
private:
int a,b;
public:
simple()
{
a=0;
b=0;
cout<<"\n Constructor of class-simple";
}
~simple() {
cout<<"\n Destructor of class-simple";
}
void getdata(int x,int y)
{
a=x;
b=y;
}
void putdata()
{
cout<<"The two integers are"<<a<<"\t"<<b;
cout<<"\n The sum of variables "<<a+b;
}
};
void main() {
simple s;
s.getdata(5,6);
s.putdata();
getch();
}
```

Output :

Constructor of class-simple

The two integers are 5 6

The sum of the variables 11

Destructor of class-simple

15. Write the Output for given C++ Program:

```
#include<iostream.h>
#include<conio.h>
class simple
{ int a,b,sum;
static int count;
public:
void assign(int i,int j)
{
a=i;
b=j;
sum = a+b;
count++;
}
void display()
{
cout<<“\n The sum of two numbers ... “<<sum;
cout<<“\n Count “<<count;
}
};
int simple::count = 0;
void main()
{
simple p1,p2,p3;
p1.assign(10,20);
p1.display();
p2.assign(20,30);
p2.display();
p3.assign(30,40);
p3.display();
}
```

Output :

The sum of two numbers 30

Count 1

The sum of two numbers 50

Count 2

The sum of two numbers 70

Count 3

16. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class simple
{
private:
int a,b;
public:
simple()
{
a=0;
b=0;
cout<<“\n Hello”;
}
~simple()
{
cout<<“\n Bye”;
}
void getdata()
{
cout<<“\n Value of a and b.”;
a=5;
b=6;
}
void putdata()
{
cout<<“The two integers are”<<a<<“\t”<<b;
cout<<“\n The sum is”<<a+b;
}
};
void main()
{
clrscr();
simple s;
s.getdata();
s.putdata();
getch();
}
```

Output :

Hello

Value of a and b :

The two integers are 5 6

The sum is 11

Bye

17. Write the Output for given C++ Program:

```
#include <iostream.h>
#include <conio.h>
class sum
{
    int a,b, s;
public:
    sum(){
        cout<<“\n Non-parameterized Constructor “;
        a=10;
        b=-20;
        s=0;
    }
    sum( int c, int d)
    {
        cout<<“\n Parameterized constructor “;
        a=c;
        b=d;
        s=0;
    }
    void addition()
    {
        s=a+b;
    }
    void putdata()
    {
        cout<<“\n The numbers are”;
        cout<<a<<“\t”<<b;
        cout<<“\n The sum\t “<< s;
    }
};
void main()
{
    clrscr();
    add obj_1,obj_2(50,100);
    obj_1.addition();
    obj_2.addition();
    cout<<“\n Object 1 : “;
    obj_1.putdata();
    cout<<“\n Object 2 : “;
    obj_2.putdata();
    getch();
}
```

Output:

Non-parameterized constructor

Parameterized constructor

Object 1 :

The numbers are 10 -20

The sum -10

Object 2 :

The numbers are 50 100

The sum 150

18. Write the Output for given C++ Program:

```
#include<iostream.h>
#include<conio.h>
class base
{
public:
    base()
    {
        cout<<“\nConstructor of base class... “;
    }
    ~base()
    {
        cout<<“\nDestructor of base class.... “;
    }
};
class derived1:public base
{
public :
    derived1()
    {
        cout << “\nConstructor of derived1 ...”;
    }
    ~derived1()
    {
        cout << “\nDestructor of derived1 ...”;
    }
};
class derived2:public base
{
public :
    derived2()
    {
        cout << “\nConstructor of derived2 ...”;
    }
    ~derived2()
    {
        cout << “\nDestructor of derived2 ...”;
    }
};
void main()
{
    derived2 x;
}
```

Output:

Constructor of base class...

Constructor of derived2 ...

Destructor of derived2 ...

Destructor of base class....

19. Write the Output for given C++ Program:

```
#include<iostream.h>
#include<iostream.h>
#include<conio.h>
class base
{
public:
base()
{
clrscr();
cout<<“\nConstructor of base class... “;
}
~base()
{
cout<<“\nDestructor of base class.... “;
}
};
class derived:public base
{
public :
derived()
{
cout << “\nConstructor of derived ...”;
}
~derived()
{
cout << “\nDestructor of derived ...”;
}
};
class derived2:public base
{
public :
derived2()
{
cout << “\nConstructor of derived2 ...”;
}
~derived2()
{
cout << “\nDestructor of derived2 ...”;
}
};
void main()
{
derived b;
derived c;
base a;
}
```

Output:

Constructor of base class...

Constructor of derived ...

Constructor of base class...

Constructor of derived ...

Destructor of base class....

Destructor of derived ...

Destructor of base class....

Destructor of derived ...

Destructor of base class....

20. Write the Output for given C++ Program:

```
#include<iostream.h>
#include<conio.h>
class base
{
public:
base()
{
cout<<“\nConstructor of base class... “;
}
~base()
{
cout<<“\nDestructor of base class.... “;
}
};
class derived:public base
{
public :
derived()
{
cout << “\nConstructor of derived ...”;
}
~derived1()
{
cout << “\nDestructor of derived ...”;
}
};
class derived2:public base
{
public :
derived2()
{
cout << “\nConstructor of derived2 ...”;
}
~derived2()
{
cout << “\nDestructor of derived2 ...”;
}
};
```

```

void main()
{
derived x1;
derived2 x;
}

```

Output:

Constructor of base class...
Constructor of derived ...
Constructor of base class...
Constructor of derived2 ...
Destructor of derived2 ...
Destructor of base class....
Destructor of derived ...
Destructor of base class....

21. Write the Output for given C++ Program:

```

#include<iostream.h>
#include<conio.h>
class sum
{
int a,b,s;
public:
sum()
{
cout<<"\nConstructor without Parameters";
a=7;
b=12;
s=0;
}
sum(int x, int y)
{
cout<<"\nParameterized Constructor";
a=x;
b=y;
s=0;
}
}

```

```

void addition()
{
s=a+b;
}
void display()
{
cout<<"\nThe Numbers are";
cout<<a<<"\t"<<b;
cout<<"\nThe Sum ="<<s;
}
};

```

Output :

Constructor without Parameters
Parameterized Constructor...
Object P
The numbers are 7 12
The Sum = 19
Object Q
The Numbers are 15 18
The Sum = 33