

CS - DEBUG PROGRAMS AND ANSWERS

1. Debug the Error for given C++ Program:

```
#include <stdio.h>
class number
{ int i;
void set()
{ i =9;
}
void display()
{
cout>>i;
}
void -operator()
{
i =-1;
};
void main()
{
number obj;
obj.set();
-obj;
obj.display();
getch();
}
```

Line No	Debug Code	Corrected Code
1	#include<stdio.h>	#include <iostream.h>
4-5	Missing member function	public:
11	cout>>i;	cout<<i;
13	void-operator()	void operator-()
15-16	Closing set bracket missing	}

2. Debug the Error for given C++ Program:

output

Feet : 12

Inches : 11

Feet : 24

Inches : 10

Feet : 36

Inches : 21

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;
cout<<"\n Inches:">>inches;
}
distance operator+[distance d2]
{
distance d3
d3.feet=feet+d2.feet;
d3.inches=inches+d2.inches;
return d3;
}
};
void main()
{
clrscr();
distance dist1,dist2;
dist1.distance_assign(12,11);
dist2.distance_assign(24,10);
distance dist3=dist1+dist2;
dist3.display();
}
```

```

dist2.display();
dist3.display()
getch();
}

```

Line No	Debug Code	Corrected Code
6	PUBLIC:	public:
9	feet=F;	feet=f;
14	cout<<"\n Feet">>feet;	cout<<"\n Feet"<<feet;
15	cout<<"\n Inches">>inches;	cout<<"\n Inches"<<inches;
17	distance operator+[distance d2]	distance operator+(distance d2)
19	distance d3	distance d3;
23]	}
34	dist3.display()	dist3.display();

3. Debug the Error for given C++ Program:

```

#include <iostream.h>
#include <conio.h>
class simple
{
private:
int a,b
public
simple()
{
a=0;
b=0;
cout>>"\nConstructor of class-simple";
}
simple()
{
cout<<"\nDestructor of class-simple";
}
void getdata()
{
Cout<<"\nEnter values for a and b";
cin<<a<<b;
}

```

```

void putdata()
{
Cout<<"\nTwo integers..."<<a<<"\t\b";
Cout<<"\n The Sum of the variables..."<<a+b;
}
};
void main()
{
simple s;
s.getdata();
s.putdata();
}

```

Output :

```

A=5
B=6
Destructor
Output:Constructor of class-simple
Enter values for a and b 6 7
Two integers ..... 6 7
The sum of the variables ..... 13
Destructor of class-simple

```

6	int a,b	int a,b;
7	public	public:
12	cout>>"\n Constructor of class-simple;	cout<<"\n Constructor of class-simple;
14	simple()	~simple()
20	Cout<<"\nEnter values for a and b";	cout<<"\nEnter values for a and b";
21	cin<<a<<b;	cin>>a>>b;
25	Cout<<"\n Two integers..."<<a<<"\t\b";	cout<<"\n Two integers..."<<a<<"\t"<<b;
26	Cout<<"\n The Sum of the variables..."<<a+b;	cout<<"\n The Sum of the variables..."<<a+b;
33	s.putdata()	s.putdata();

4. Debug the Error for given C++ Program:

```
#include <stdio.h>
class a
{ int i;
public:
A()
{i =0;}
A(X)
{ x=i; }
A(AX)
{i =x.i;
};
void output()
{
cout<<>>"i="<>>i<>>
}
};
void main
{
A a1(6), a2(a1)
a1.output();
a2.output {}.
```

Output :

```
i=6
i=6
```

1	#include<stdio.h>	#include<iostream.h>
2	class a	class A
9	A(X)	A(int x)
11	A(AX)	A(A x)
13	i=x.i;	i=x.i;
14	};	}
17	cout<>>>"i="<>>i<>>	cout<<<<"i="<<<i;
20	void main	void main()
22	A a1(6),a2(a1)	A a1(6),a2(a1);
24	a2.output{ }.	a2.output();
25	} is missing	}

5. Debug the Error for given C++ Program:

#include <iostream.h>	~~simple()
#include <conio.h>	{
class simple	cout<<<<"\n Destructor of class-simple";
{	}
private:	void generate();
int a,b;	{
public:	cout<<<<"\n Enter value for a and b";
simple[]	cin a>>b;
{	}
a=0;	void putdata()
b=0;	{
cout<<<<"\n Constructor of class-simple";	cout<<<<"\n Two integers..."<<<a<<<"\t" b;
}	cout<<<<"\n The sum of the variables..."<<<a+b;
	}
	};

```

void main()
{
simple S;
S.getdata();
S.putdata();
}
}

```

Output :

```

Constructor of class-simple
Enter values for a and b 5 6
Two integers ... 5 6
The sum of the variables... 11
Destructor of class-simple

```

```

8   simple[ ]
12  cout<<"\n constructor of class-simple";
14  ~simple()
18  void generate();
20  cout<<"\n Enter values for a and b";
21  cin >> a >> b;
25  cout<<"\n Two integer..."<<a<<"\t" << b;
35  } is extra to be remove

```

```

simple()
cout<<"\n constructor of class-simple";
~simple()
void getdata()
cout<<"\n Enter values for a and b";
cin >> a >> b;
cout<<"\n Two integer..."<<a<<"\t" << b;

```

6 . Debug the Error for given C++ Program:

```

#include iostream.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 get();
{
cout<<"\n Enter value for a and b ...";

```

```

cin<<a<<b; }
void data()
{
cout<<"\n The two integers..."<<a<<"\t" < b;
cout<<"\n The sum of the variables..."<<a+b; }
};
void ()
{
simp s;
s.getdata();
s.putdata();
} }

```

Output :

```

Constructor of class-simple
Enter values for a and b
2
4
Two integers ... 2 4
The sum of the variables... 6
Destructor of class-simple

```

```

1   #include iostream.h]
2   class simple()
4   private;
11  cout<<"\n constructor of class-simple";
13  !simple()
15  Cout<<"\n Destructor of class-simple";
17  void get()
19  cin<<a<<b;
20  void data()
25  void ()
26  simp s;
29  } } (one of } must be delete. It's Extra)

```

```

#include <iostream.h>
class simple
private:
cout<<"\n constructor of class-simple";
~simple()
cout<<"\n Destructor of class-simple";
void getdata()
cin >> a >> b;
void putdata()
void main()
simple s;
}

```

7 . Debug the Error for given C++ Program:

```
# include<oistream.h>
#include<conio.h>
class add
{ int num1, num2, sum;
public
add()
{ cout<<“\n Constructor without parameters.. “;
num1= 0;
num2= 0;
sum = 0;
}
ad ( int s1, int s2 )
{
cout<<“\n parameterized constructor.. “;
num1=S1;
num2=S2;
sum=NULL;
}
add ( add a )
{
cout<<“\n Copy constructor... “;
num1= a.num1;
num2=a.num2;
sum=NULL;
}
void getdata()
{
num1=10;
num2=20;
}
```

```
void putdata()
{ cout<<“\n The numbers are...”;
cout<<num1<<“\t”<num2;
sum=num1+num;
cout<<“\n The sum of the numbers “<< sum;
}
void main
{
add ab (10, 20), c(b);
a.getdata();
cout>>“\n Object a : “;
a.putdata();
cout>>“\n Object b : “;
b.putdata();
cout>>“\n Object c : “;
c.putdata();
}
```

Output :

```
-----
Constructor without parameters..
parameterized constructor..
Copy constructor...
Object a :
The numbers are.. 10 20
The sum of the numbers 30
Object b :
The numbers are.. 10 20
The sum of the numbers 30
Object c :
The numbers are.. 10 20
The sum of the numbers 30
```

```
1      #include <oistream.h>
4      public
11     ad(int s1,int s2)
23     void getdata()
30     cout<<num1<<“\t”<num2;
32     sum=num1+num;
33     Missing statement };
34     void main
36     add ab(10,20),c(b);
38     cout>>“\n Object a : “;
40     cout>>“\n Object b : “;
42     cout>>“\n Object c : “;
```

```
#include <iostream.h>
public:
add(int S1,int S2)
Void getdata()
cout<<num1<<“\t”<<num2;
sum=sum1+sum2;
};
void main()
add a,b(10,20),c(b);
cout<<“\n Object a : “;
cout<<“\n Object b : “;
cout<<“\n Object c : “;
```

8 . Debug the Error for given C++ Program:

```
# include<iostream.h>
#include<conio.h>
class sum()
{
private:
int a=0,b=0;
public:
sum();
{
a=b=5;
cout>>"\n Constructor ";
}
sum();
{
cout>>"\n Destructor ";
}
```

```
void inputdata ( int n1, int n2 )
{
cout<<"\n parameterized constructor.. ";
a=n1;
b=n2;
}
display()
{ cout<<"\n The numbers are..."<<a<<b;
cout<<"\n The sum ="<<a+b;
}
};
void main()
{
sum s;
s.inputdata(10);
s.display(20);
};
```

```
3     class sum()
6     int a=b=0;
8     sum();
11    cout>>"\nConstructor";
13    sum();
15    cout>>"\nDestructor";
25    cout<<"\n The sum ="<<a+b;
32    s.inputdata(10);
33    s.display(20);
34    }; ( semi-colon ; is not allowed)
```

```
class sum
int a,b;
sum()
cout<<"\n Constructor";
~sum()
cout<<"\n Destructor";
cout<<"\n The sum ="<<a+b;
s.inputdata(10,20);
s.display();
}
```

9 . Debug the Error for given C++ Program:

```
@include <iostream.h>
class add
{ int sum;
protected:
int num1,num2;
public:
void add();
{
num1=num2=sum=0;
cout<<"\n Add Constructor;
}
void accept();
{
num1=12;
num2=14;
}
void plus();
{
sum=num1+num2;
cout<<sum;
```

```
}
class subtrac()
{ int sub;
public:
void subtract();
{
sub=0;
cout<<"\n Subtract constructor
void minus();
{
add:accept();
sub=num1-num2;
cout<<sub;
};
void main()
{
subtract s;
s.accept;
s.plus;
s.minus;
}
```

1	@include <iostream.h>	#include <iostream.h>
8	void add();	void add()
11	cout<<"\n Add Constructor;	Cout<<" \n Add Constructor";
13	void accept();	void accept()
18	void plus();	void plus()
23	class subtrac()	class subtract:public add
27	void subtract();	void subtract()
30	cout<<"\n Subtract constructor	cout<<"\n Subtract constructor";
30-31	} (is missing)	}
31	void minus();	void minus()
33	add:accept();	add::accept();
35-36	} (is missing)	}
40	s.accept;	s.accept();
41	s.plus;	s.plus();
42	s.minus;	s.minus();

10 . Debug the Error for given C++ Program:

```
#include [iostream.h]
#include conio.h
class simple()
[
PRIVATE:
int ab;
public:
simple[]
{
a=0;
b=0;
cout>>"\N Constructor of class-simple";
}
!simple()
{
Cout<<"\n Destructor of class-simple";
}
void getdata[]
{
cout<<"\n Enter values for a and b ...";
```

```
cout<<a<<b;
}
void putdata()
{
cout<<"\N The two integers..."<<a,<<"\t"<< b;
cout<<"\n The sum of the variables..."<<a+b;
}
}
void MAIN ()
{
Simp s;
s.getdata();
s.putdata();
}
```

Output :

```
Constructor of class-simple
Enter values for a and b 2 4
Two integers ... 2 4
The sum of the variables... 6
Destructor of class-simple
```

1	#include [iostream.h]	#include <iostream.h>
2	#include conio.h	#include <conio.h>
3	class simple()	class simple
4	[{
5	PRIVATE:	private:
6	int ab;	int a,b;
8	simple[]	simple()
12	cout>>"\N Constructor of class-simple";	cout<<"\n Constructor of class-simple";
14	!simple()	~simple()
18	void getdata[]	void getdata()
21	cout<<a<<b;	cin>>a>>b;
25	cout<<"\N The two integers..."<<a,<<"\t"<< b;	cout<<"\n The two integers..."<<a<<"\t"<< b;
28	}	};
29	void MAIN()	void main()
31	Simp s;	simple s;

11 . Debug the Error for given C++ Program:

output:

```
3
4
5
#include <iostream.h>
class A
{
private
int a;
public;
int a2;
void getdata{ }
{
a1=3;
a2=5 a3=5;
}

4     private
5     int a;
6     public;
8     void getdata{ }
11    a2=5 a3=5;
13    protected
15    }
16    class B:: public A()
21    int b1;b2;b3;
22    getdata{ };
26    cout>>b1>>b2>>b3;
28    }
29    void main
32    d.func();
33    der.b3=0;
```

```
protected
int a3;
}
class B::public A()
{
public:
void func()
{ int b1;b2;b3;
getdata{ };
b1=a1;
b2=a2;
a3=b3;
cout>>b1>>b2>>b3;
}}
void main
{
B der;
d.func();
der.b3=0;
}
```

```
private:
int a1;
public:
void getdata()
a2=5; a3=5;
protected:
};
class B:: public A
int b1,b2,b3;
getdata();
cout<<b1<<b2<<b3;
};
void main()
Der.func();
der.a3=0;
```

12 . Debug the Error for given C++ Program:

output:

```
Constructor
Derived Constructor
Derived Destructor
Destructor
#include <iostream.h>
#include <conio.h>
class base
{
public
base
{
cout>>"\n Constructor";
}
```

```
base()
{
cout<<"\n Destructor";
}
};
class derived? public base
{
public:
derived()
{
cout<<"\n Derived Constructor";
}
~derived()
{
cout<<"\n Derived Destructor";
}
};
```



```
void main()
{
derived x
}
```

Line No	Debug Code	Corrected Code
5	public	public:
6	base	base()
8	cout>>”\n Constructor”;	cout<<”\n Constructor”;
10	base()	~base()
15	class derived? public base	class derived:public base
20	cout<<”\n Derived Destructor”:	cout<<”\n Derived Destructor”;
27	void main	void main()
29	derived x	derived x;

13 . Debug the Error 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 baseclass
{
```

```
public :
Derived()
{
cout << “\nConstructor of derived ...”;
}
^derived()
{
cout << “\nDestructor of derived ...”;
}
};
void MAIN()
{
derived x(5);
}
```

Output:

```
-----
Constructor of base class...
Constructor of derived ...
Destructor of derived ...
Destructor of base class....
```

Line No	Debug Code	Corrected Code
1	\$include<iostream.h>	#include<iostream.h>
2	#include conio.h	include <conio.h>
5	PUBLIC:	public:
6	base[]	base()
8	cout>>”\N constructor of base class”;	cout<<”\n constructor of base class”;
10	^base()	~base()
12	Cout<<”\nDestructor of base class;	cout<<”\nDestructor of base class”;
15	class::public baseclass	class:public base
18	Derived()	derived()
22	^derived()	~derived()
27	void MAIN()	void main()
29	derived x(5);	derived x;

14. Debug the Error for given C++ Program:

```
#include <iostream.h>
class A
{
private:
int a1;
public:
int a2;
protected:
int a3;
};
class B:public A()
{
```

```
public:
void func()
{ int b1b2b3;
b1=a1;
b2=a2;
b3=a3;
};
void Main()
{
B der;
der.a3=0;
der.func();
}
```

Line No	Debug Code	Corrected Code
16	int b1 b2 b3;	int b1,b2,b3;
17	b1=a1;	a1 should be declared under protected or public
22	void Main()	void main()
25	der.a3=0;	a3 should be declared under public to access it.
28	}	should be deleted

15. Debug the Error for given C++ Program:

```
#include <iostream.h>
class base
{
base()
{
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 derived:publicly derived2
{
public:
derived2()
{
cout<<”\nConstructor of derved2”;
}
! derived2()
{
cout<<”\n Destructor of derived2”;
}
};
void base()
{
x.derived2;
x1.derived;
};
```

Line No	Debug Code	Corrected Code
8	;base()	~base()
12	}	};
13	class derived::public-base	Class derived:public base
15	public	public:
20	!derived()	~derived()

23	};	}
25	class derived:publicly derived2	class derived2:public derived
32	! derived2()	~derived2()
37	void base()	void main()
39	x.derived2;	derived2 x;
40	x1.derive;	derived x1;
41	};	}

16. Debug the Error for given C++ Program:

```
#include <isostream.h>
class base
{
public
base()
{
cout>>"\nConstructor ";
}
+base()
{
cout>>"\nDestructor";
};
class derived::public-base
{
public
der1()
{
cin << "\nConstructor";
}
};
void main()
{
der.obj;
obj.der1();
}
```

Line No	Debug Code	Corrected Code
1	#include<isostream.h>	#include<iostream.h>
4	public	public:
7	cout>>"\nConstructor";	cout<<"\nConstructor";
9	+base()	~base()
11	cout>>"\nDestructor";	cout<<"\nDestructor";
11-12	}is missing	}
13	class der::public base	class der:public base
15	public	public:
16	der1()	der()
18	cin << "\nConstructor";	cout << "\nConstructor";
23	der.obj;	der obj;
24	obj.der1();	der obj1;

17. Debug the Error for given C++ Program:

```
class add
(
int s=0;
protected;
int n1 n2;
public:
add[];
{
n1=n2=0;
}
accept()
{
cin>>n1>>n2;
}
plus()
{
s=n1+n2;
}
};
```

```
class add:privetly subtract
{
sub;
subtract()
{
sub=0;
minus()
{
add::acc();
sub=n1-n2;
cout<<sub;
};
}
void main[]
{
obj subtract;
obj.minus();
}
```

Line No	Debug Code	Corrected Code
2	({
4	protected;	protected:
5	int n1 n2;	int n1,n2;
7	add[]	add()
20	class add:privetly subtract	class subtract:private add
22	sub;	int sub;
25	sub=0	sub=0;
26-27	}is missing	}
28	add::acc();	add::accept();
31	};	}
32	}	};
33	void main[]	void main()
35	obj subtract;	subtract obj;
36	obj.minus();	obj.minus();

18. Debug the Error for given C++ Program:

output

Concatenated Strings.....testrun

Program:

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

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

```
#include<string.s>
```

```
class strings
```

```
{
```

```
Char s[10];
```

```
Public;
```

```
Strings()
```

```
{
```

```
S[0]="\0";
```

```
}
```

```
string (char *c)
```

```
{
```

```
strcpy(s,c);
```

```
}
```

```
Char * operator – (strings x1)
```

```
{
```

```
char *temp;
```

```
strcpy(temp,s);
```

```
strcat(temp,x1,s);
```

```
return temp;
```

```
};
```

```
void main[]
```

```
{
```

```
clrscr();
```

```
strings s1("test"),s2("run\0");
```

```
char *concatstr;
```

```
concatstr=s1+s2;
```

```
cout<<>>"\nconcatenated strings....">>concatstr;
```

```
}
```

Line No	Debug Code	Corrected Code
3	#include<string.s>	#include<string.h>
6	Char s[10];	char s[10];
7	Public:	public:
8	Strings()	strings()
10	S[10]="\0";	s[10]="\0";
12	string(char*c)	strings(char*c)
14	strcpy(s,c);	strcpy(s,c);
16	Char * operator-(strings x1)	char operator-(strings x1)
19	strcpy(temp,s);	strcpy(temp,s);
20	strcpy(temp,x1,s);	strcpy(temp,x1.s);
21-22	} is missing	}
23	void main[]	void main()
25	clrscr();	clrscr();
26	strings s1("test"), s2("run\0");	strings s1("test"), s2("run\0");
29	cout<>>"\n concatenate strings...">>concatstr;	cout<<>>"\n Concatenate Strings..."<<concatstr;