

# Assignment\_2(OOP in CPP, Class & Objects, Constructor & Destructor)

1. Differentiate between Data Abstraction and Data Hiding with examples.
2. Write the output of the C++ program code:-

```
class Eval
{
    Char Level; int Point;
public:
    Eval()
    { Level = 'E';
      Point = 0; }
```

```
void Sink(int L)
    { Level -= L; }
void Float(int L)
    { Level += L;
      Point++; }
void show()
    {cout<<Level<<"#"<<Point<<endl; } };
```

3. Rewrite the following program after removing the syntax errors, underline each correction:-

```
#include<iostream.h>
Class Transport
{ char Model[20];
  char Name[20];
  void Get() { gets(Model);
              gets(name); }
  void Show { cout<<Model<<endl;
```

```
puts(Name); }
};
Void main()
{ transport T;
  Get();
  Show();
```

4. Find the output of the following C++ program:-

```
#include<iostream.h>
class Train
{ int Tno, Tripno, Personcount;
public:
  Train(int Tmno = 1)
  { Tno=Tmno; Tripno = 0;
    Personcount=0; }
  void Trip( int Pc = 20)
  { Tripno++; Personcount += Pc; }
  void Show()
  { cout<<Tno<<":"<<Tripno<<":"<<Personcount<<endl; } };
```

```
void main()
{
  Train M(5), T;
  M.Trip();
  T.Trip(50);
  M.Show();
  M.Trip(30);
  T.Show();
  M.show();
}
```

5. Answer the questions (i) and (ii) after going through the following class:-

```
class Race
{
  int carno, Track;
public:
  Race(); //Function 1
  Race(int CN); //Function 2
  Race(Race &M); //Function 3
  void Register (); //Function 4
  void Drive(); //Function 5
```

```
};
void main()
{
  Race R;
  :
  :
  :
}
```

- (i.) Out of the following, which of the option is correct for calling function 2?

- Option 1 : Race T(30);
- Option 2 : Race U(R);

(ii.) Name the feature of OOP, Which is illustrated by Function 1, Function 2 and Function 3 combined together.

6. Rewrite the following program after removing the syntax errors, underline each correction:-

```
#include<iostream.h>
#include<stdio.h>
Class MyStudent
{
    int StudentId = 101;
    char Name[20];
public
    Mystudent() {}
void Register()
    { cin>>StudentId; gets(Name); }
```

```
void Display()
    { cout<<StudentId<<":"<<Name<<endl; }
};
void main()
{
    MyStudent MS;
    Register.MS();
    MS.Display();
}
```

7. Find the output of the following code:-

```
#include<iostream.h>
class myclass
{
    int a, b;
public:
    void set(int I, int J)
        { a = I; b = J; }
    void show()
        { cout<<a<<" "<<b<<'\n'; }
};
```

```
int main()
{
    myclass o1, o2;
    o1.set(10, 4);
    o2 = o1;
    o1.show();
    o2.show();
    return 0;
}
```

8. Rewrite the following program after removing the syntax errors, underline each correction:-

```
#include<iostream.h>
class myclass
{
    int a, b;
public:
    void set(int I, int J) { a = I; b = J; }
    void show() { cout<<a<<" "<<b<<'\n'; }
};
class yourclass
{
    int a, b;
public:
    void set(int I, int J) { a = I; b = J; }
```

```
void show() { cout<<a<<" "<<b<<'\n'; }
};
int main()
{
    myclass o1;
    yourclass o2;
    o1.set(10, 4);
    o2 = o1;
    o1.show();
    o2.show();
    return 0;
}
```

9. Describe the two ways of defining the member functions:-

10. Define a class named HOTEL in C++ with the following descriptions:-

Private members:

- Rno //to store Room No.

- Name //to store Customer Name
- Tariff //to store per day Charges
- NOD //to store Number of Days to Stay
- Calc() //to calculate and return Amount as NOD\*Tariff

Public members:

- Checkin() //to enter the Rno, Name, Tariff and NOD
- Checkout() //to display Rno, Name, Tariff, NOD and Amount to be displayed by calling function Calc();

11. What is the output of the following program:-

<pre>#include&lt;iostream.h&gt; class Item { static int count; int number; public:     void Get(int a) {         number = a;         count++; } void Show() {     cout&lt;&lt;"count:"&lt;&lt;count&lt;&lt;"\n"; } }; void main() {</pre>	<pre>Item a, b, c; a.Show(); b.Show(); c.Show(); a.Get(100); b.Get(200); c.Get(300); cout&lt;&lt;"After Reading Data" &lt;&lt;"\n"; a.Show(); b.Show(); c.Show(); }</pre>
---	---

12. Define Constructor. What is its need? What are the different types of constructors? Explain with examples.

13. What is Default Constructor? How does it differ from destructor?

14. What do you mean by the term "Constructor Overloading"? Give examples.

15. What is Copy Constructor? Give an example in C++ to illustrate Copy Constructor.

16. What do you mean by static variables and static function in a class? Give examples.

17. Answer the questions (i) and (iii) after going through the following class:-

```
class Exam
{
    int year;
public:
    Exam(int y) //Constructor 1
    {
        Year = y; }
    Exam(Exam &t); //Constructor 2
};
```

(i.) Create an object, such that it invokes Constructor 1.

(ii.) Write the complete definition for Constructor 2.

(iii.) Distinguish between the two statements:-

```
Exam T1(10);
Exam T2 = Exam(20);
```

18. Answer the questions (i) and (ii) after going through the following class:-

```

class Maths
{
    char Chapter[20];
    int marks;
public:
    Maths()
    //Member Function 1
    {
        strcpy(Chapter,"Geometry");
        Marks=10;
    }
    ~Maths()
    //Member Function 2
    {
        cout<<"Chapter Over";
    }
};
    cout<<"Chapter Initialized";
}

```

(i.) Name the specific features of class shown by Member Function 1 and Member Function 2 in the above example.

(ii.) How would Member function 1 and Member Function 2 get executed?

19. What is the output of the program:-

```

class ABC
{
    int p;
public:
    ABC() { p=10;
    cout<<"Hello"<<p<<endl; }
    ~ABC() { cout<<"World"<<endl; }
};
void main()
{
    ABC w, x, y, z;
    cout<<"BLOCK 1"<<endl;
    ABC m;
    cout<<"BLOCK 2"<<endl;
    ABC n;
    cout<<"BLOCK 3"<<endl;
}

```

20. Define a class TravelPlan in C++ with the following descriptions:-

Private members:

- PlanCode of type long
- Place of type character array(string)
- Number\_of\_travellers of type integer
- Number\_of\_buses of type integer

Public members:

- A constructor to assign values of PlanCode as 1001, Place as "Agra", Number\_of\_travellers as 5, Number\_of\_buses as 1.
- A function NewPlan() which allows user to enter PlanCode, Place and Number\_of\_travellers..
- Also, assign the value of Number\_of\_buses as per the following conditions:

Number_of_travellers	Number_of_buses
Less than 20	1
Equal to or more than 20 and less than 40	2
Equal to 40 or more than 40	3

- A function ShowPlan() to display the content of all the data members on screen.

21. When will you make a function inline and why?

22. What is the difference between a struct and a class in C++?

. . .

## Set-1 (C++ BASICS) Answer\_Sheet

- The parameters in the function call statement (or calling function) are called as Actual Parameters.  
The parameters in the function definition (or called function) are called as Formal Parameters.  
Eg:  

```
void manip(int x, int y)
{ ---
---
}
void main( )
{
int a,b;
----
Manip(a,b);
}
```

Here a,b are Actual Parameters and x,y are Formal Parameters.
- (i) iomanip.h (ii) math.h
- ```
#include <iostream.h>
#include <stdio.h>
class MyStudent
{ int StudentId;
char Name[20];
public:
MyStudent( ) { StudentId=1001; }
void Register( )
{ cin>>StudentId;
gets(Name);
}
void Display( )
{ cout<<StudentId<<":"<<Name<<endl;
}
};
void main( )
{ MyStudent MS;
MS.Register();
```
- MS.Display( );
- teRmttoe
- (iv) 51#50#
- iostream.h, iomanip.h
- ```
#include<iostream.h>
void Jumpto(int N1, int N2);
void main( )
{ int First = 10, Second = 20;
Jumpto(First,Second);
Jumpto(Second);
}
void Jumpto(int N1, int N2 = 20)
{ N1=N1+N2;
cout<<N1<<N2;
}
```
- Nnd@\*ork!\*
- [1] 10 & 15  
[2] 21 & 13  
[1] 11 & 16  
[2] 22 & 14
- (ii) CHN: KOL : CHN:  
(iv) KOL : CHN : KOL
- iostream.h, iomanip.h
- ```
#include<iostream.h>
void Callme(int Arg1,int Arg2);
void main( )
{ int One=10,Two=20;
Callme(One;Two);
Callme(Two);
}
void Callme(int Arg1,int Arg2=1)
{ Arg1=Arg1+Arg2;
cout<<Arg1<<Arg2; }
```
- hat@\*PVUQVU\*
- Line1 5 & 8  
Line2 11 & 9  
Line1 6 & 9  
Line2 12 & 10

15. (i) SOUTH : EAST : SOUTH :

(iv) SOUTH : EAST : EAST :

16. **Run Time Errors:** Errors that occur during the execution of a program are called as run time errors. It is caused of some illegal operation taking place or unavailability of desired or required conditions for the execution of the program. For instance, if enough memory is not available or an expression is trying to divide a number by zero are run-time errors.

Eg: Division by zero.  $c=a/b$ ;

User will give the values of a and b at the time of program execution. If he gives the value of b as '0' , then division by zero, i.e. a run time error occurs.

**Syntax Errors:** Syntax errors occur when rules of a programming languages (syntax) is misused. Ie when a grammatical rule of C++ is violated.

Eg:  $c=a+b$

In this statement,since there is no semicolon at the end of the statement, there will occur a syntax error.

17. string.h, stdio.h

18. #include<iostream.h>

```
const int Max = 10;
```

```
void main()
```

```
{ int Numbers[] = {20,50,10,30,40};
```

```
for(int Loc=Max-1;Loc>=10;Loc--)
```

```
cout<<Numbers[Loc];}
```

19. 20 25 30

20 25 30

Number=30

20. (ii)94