Assignment - 2

Q1⇒ How we can define a function inline, what are the benefits of this methodology?

Ans⇒ **Inline Function** :- It is a powerful concept that is commonly used with classes. If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time.

Inline Function Benefits are :-

1.) It saves the overhead of push / pop variables on the stack when function is called.

2.) It saves overhead of a return call from a function.

3.) When you inline a function, you may enable compiler to perform context specific optimization on the body of function.

Q2⇒ What is Constructor and Destructor? Explain various types of constructor. It is mandatory to use constructor in a class. Justify it?

Teacher's Signature.....

Ans → **Constructor** :- They are special class functions which performs initialization of every object. The compiler calls the constructor whenever an object is created.

Destructor :- It is used to destroy the class object.

Types of Constructor :-

1. **Default constructor** :- It is the constructor which doesn't take any argument. It has no parameter.

Syntax :-
```
class name ( parameter1, parameter2 ...
{
    // Constructor Defination
}
```

Example :-
```
class cube
{
public:
int side ();
cube ()
{
```

Teacher's Signature.....

```
        side = 10;
    }
};
int main()
{
    Cube c;
    cout << c.side;
}
```

→ In this case, as soon as the project is created the constructor is called with initializes its data members

2 → Parameterized Constructor - These are the constructors with parameters Using this Constructor you can provide different values to data members of different objects, by passing the appropriate values as argument.

Example:-

```
Class Cube
{
    public:
        int side;
    Cube(int x)
    {
        side = x;
    }
};
```

Teacher's Signature.....

```
int main ()
{
    cube c1(10);
    cube c2(20);
    cube c3(30);
    cout << c1.side;
    cout << c2.side;
    cout << c3.side;
}
```

By using parametesized constructor in above case, we have initialized 3 objects with user defined values. We can have any number of parameters in a constructor.

3. Copy Constructor -

These are special type of constructors which is used to copy values of data members of one object into other object.

Example:-

```
class cube
{
public:
    int side;
    cube (int x)
    {
        side = x;
    }
};
```

Teacher's Signature.....

```
int main ()
{
    cube c1(c2);
    cube c2(c1);
    cout << c1.side << c2.side;
}
```

Q4→ What is mean by the following terms?

(a) Constructor Overloading

(b) Array of Object

– Constructor Overloading :–

Just like other member functions, constructors can also be overloaded. Infact when you have both default and parameterized constructors defined in your class you are overloaded constructors, one with no parameter and other with parameter.

You can have any number of constructors in a class that differ in parameter list.

Example :–

```
class Student
{
    public:
        int roll no ;
```

Teacher's Signature.....

```
String name;
Student (int x)
  {
     roll no = x;
     name = "None";
  }
Student (int x, string str)
  {
     roll no = x;
     name = str;
  }
};

int main ()
  {
     Student A(10);
     Student B(11, "John");
  }
```

- Array of Object :-

- Like array of other user-defined data types, an array of type class can also be created.

- The array of type class contains the objects of the ~~class~~ class as its individual elements.

Teacher's Signature.....

- Thus, an array of a class type is also known as an array of objects.
- An array of objects is declared in the same way as an array of any built-in data type.

Syntax:

```
class_name array_name[size];
```

Example:-

```
#include<iostream.h>
using namespace std;

class MyClass
  {
    int x
    public:
      void setX(int i)
        {
          x=i;
        }
      int getX()
        {
          return x;
        }
  };
```

```
void main()
{
    My class obs[4];
    int i;
    for(i=0; i<4; i++)
    {
        obs[i].setx(i);
    }

    for(i=0; i<4; i++)
    {
        cout<<"obs["<<i<<"].getx():"
        <<obs[i].getx()<<"\n";
    }
    getch();
}
```

Q5⇒ What is Friend Function? Write a program?

Ans→ A friend function of a class is defined outside that class scope but it has the right to access all private and protected members of the class. Even though the prototypes for friend function appears in class definations, friends are not member functions.

Teacher's Signature.....

Example:

```
#include <iostream>
using namespace std;
class Box
{
    double width;
public:
    friend void print width (Box box);
    void set Width (double wid);
};

void Box :: set width (double wid)
{
    width = wid;
}

void print width (Box box)
{
    cout << "Width of box :" << box.width << endl;
}

int main()
{
    Box box;
    box.setWidth (10.0);
    printWidth (box);
```

Teacher's Signature.....

Date
Page 26

```
    return 0;
}
```