1.

Definition

OOP stands for Object-oriented programming and is a programming approach that focuses on data rather than the algorithm, whereas POP, short for Procedure-oriented programming, focuses on procedural abstractions.

Programs

In OOP, the program is divided into small chunks called objects which are instances of classes, whereas in POP, the main program is divided into small parts based on the functions.

Accessing Mode

Three accessing modes are used in OOP to access attributes or functions – ‘Private’, ‘Public’, and ‘Protected’. In POP, on the other hand, no such accessing mode is required to access attributes or functions of a particular program.

Focus

The main focus is on the data associated with the program in case of OOP while POP relies on functions or algorithms of the program.

Data Control

In OOP, the data and functions of an object act like a single entity so accessibility is limited to the member functions of the same class. In POP, on the other hand, data can move freely because each function contains different data.

Security

OOP is more secure than POP, thanks to the data hiding feature which limits the access of data to the member function of the same class, while there is no such way of data hiding in POP, thus making it less secure.

Ease of Modification

New data objects can be created easily from existing objects making object-oriented programs easy to modify, while there's no simple process to add data in POP, at least not without revising the whole program.

Process

OOP follows a bottom-up approach for designing a program, while POP takes a top-down approach to design a program.

2. Advantages of C++ over C are as follows:

- C++ is object-oriented and it is related to real world objects, while C is procedural-oriented so it focuses on procedural abstractions.
- C++ can also use call by reference value while we can only simulate call by reference using pointers in C.
- C++ follows a bottom to top approch while C uses a top to bottom approach.
- C++ has classes while C only has structures.
- C++ accesibilty is private by default due to data abstraction.
- C++ use inheritance while C does not.
- Overloading is allowed in C++, but does not allowed in C.

- Exception Handling is allowed in C++, but does not allowed in C.

3. There are three types of constructors in C++:

- **Default constructor** is the constructor which doesn’t take any argument. It has no parameters.

- **Parameterized Constructors:** It is possible to pass arguments to constructors. Typically, these arguments help initialize an object when it is created. To create a parameterized constructor, simply add parameters to it the way you would to any other function. When you define the constructor’s body, use the parameters to initialize the object.

- **Copy Constructor:** A copy constructor is a member function which initializes an object using another object of the same class.

4. Destructor is a member function which destructs or deletes an object.

A destructor function is called automatically when the object goes out of scope:

(1) the function ends

(2) the program ends

(3) a block containing local variables ends

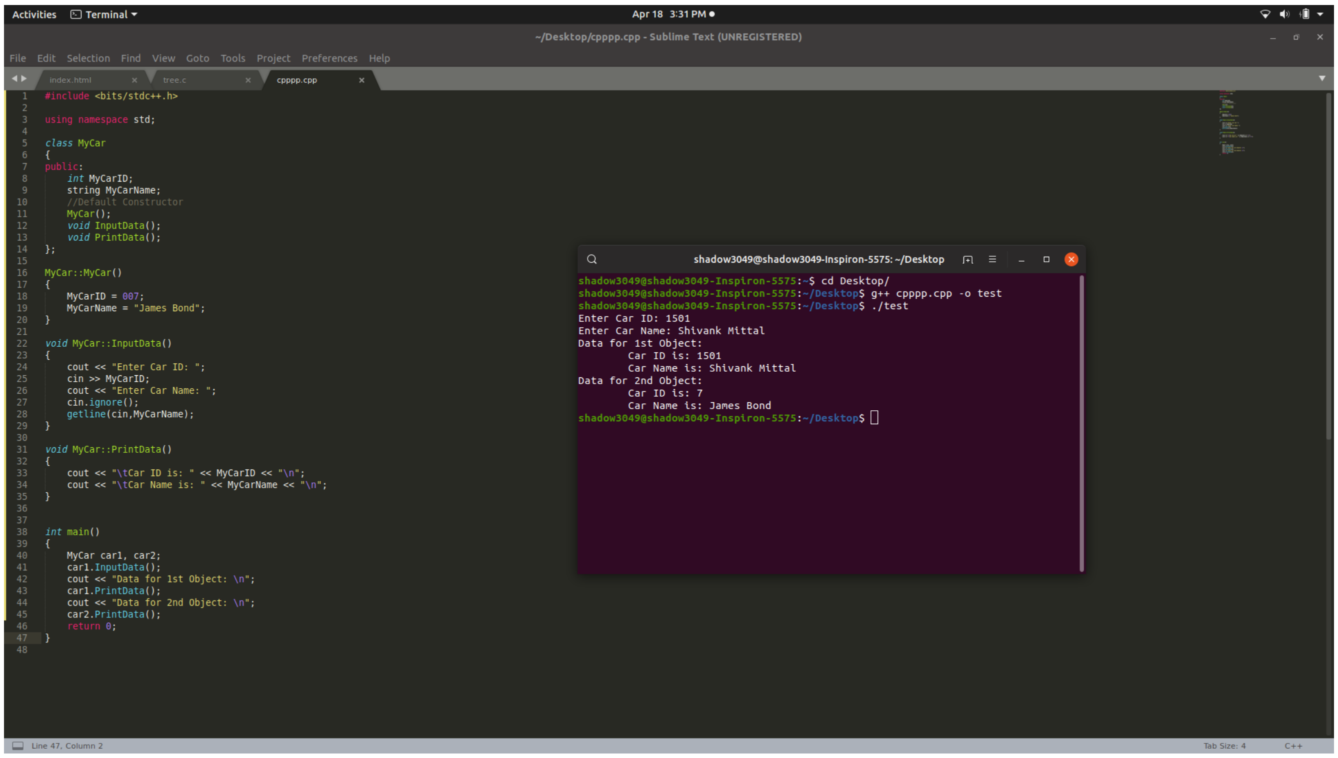
(4) a delete operator is called

Destructors have same name as the class preceded by a tilde (~).
Destructors don’t take any argument and don’t return anything.

5.

```
#include <bits/stdc++.h>


using namespace std;


class MyCar
```

```
{
public:
	int MyCarID;
	string MyCarName;
	//Default Constructor
	MyCar();
	void InputData();
	void PrintData();
};

MyCar::MyCar()
{
	MyCarID = 007;
	MyCarName = "James Bond";
}

void MyCar::InputData()
{
	cout << "Enter Car ID: ";
	cin >> MyCarID;
	cout << "Enter Car Name: ";
	cin.ignore();
	getline(cin,MyCarName);
}

void MyCar::PrintData()
{
        cout << "\tCar ID is: " << MyCarID << "\n";
        cout << "\tCar Name is: " << MyCarName << "\n";
}



int main()
{
        MyCar car1, car2;
        car1.InputData();
        cout << "Data for 1st Object: \n";
        car1.PrintData();
        cout << "Data for 2nd Object: \n";
        car2.PrintData();
        return 0;
}
```


Activities
Terminal
Apr 18 3:31 PM
~/Desktop/cpppp.cpp - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
tree.c
cpppp.cpp
shadow3049@shadow3049-Inspiron-5575: ~/Desktop
Line 47, Column 2
Tab Size: 4
C++

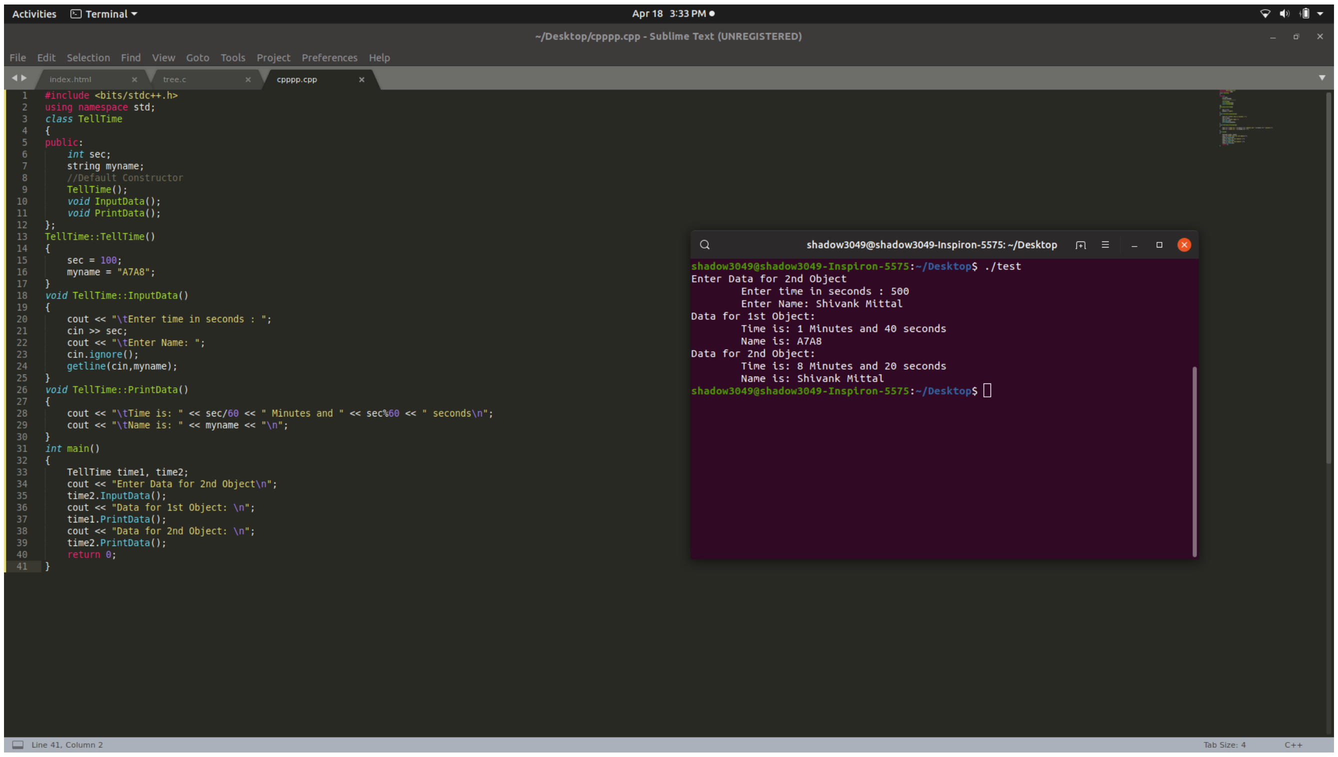
```
#include <bits/stdc++.h>

using namespace std;

class MyCar
{
public:
    int MyCarID;
    string MyCarName;
    //Default Constructor
    MyCar();
    void InputData();
    void PrintData();
};

MyCar::MyCar()
{
    MyCarID = 007;
    MyCarName = "James Bond";
}

void MyCar::InputData()
{
    cout << "Enter Car ID: ";
    cin >> MyCarID;
    cout << "Enter Car Name: ";
    cin.ignore();
    getline(cin,MyCarName);
}

void MyCar::PrintData()
{
    cout << "\tCar ID is: " << MyCarID << "\n";
    cout << "\tCar Name is: " << MyCarName << "\n";
}


int main()
{
    MyCar car1, car2;
    car1.InputData();
    cout << "Data for 1st Object: \n";
    car1.PrintData();
    cout << "Data for 2nd Object: \n";
    car2.PrintData();
    return 0;
}

```

```
shadow3049@shadow3049-Inspiron-5575:~$ cd Desktop/
shadow3049@shadow3049-Inspiron-5575:~/Desktop$ g++ cpppp.cpp -o test
shadow3049@shadow3049-Inspiron-5575:~/Desktop$ ./test
Enter Car ID: 1501
Enter Car Name: Shivank Mittal
Data for 1st Object:
        Car ID is: 1501
        Car Name is: Shivank Mittal
Data for 2nd Object:
        Car ID is: 7
        Car Name is: James Bond
shadow3049@shadow3049-Inspiron-5575:~/Desktop$ 
```

6.

```
#include <bits/stdc++.h>

using namespace std;

class TellTime

{

public:

        int sec;

        string myname;

        //Default Constructor

        TellTime();

        void InputData();

        void PrintData();

};

TellTime::TellTime()

{

        sec = 100;

        myname = "A7A8";

}

void TellTime::InputData()

{

        cout << "\tEnter time in seconds : ";

        cin >> sec;

        cout << "\tEnter Name: ";

        cin.ignore();

        getline(cin,myname);

}

void TellTime::PrintData()
```

```
{
        cout << "\tTime is: " << sec/60 << " Minutes and " << sec%60 << " seconds\n";
        cout << "\tName is: " << myname << "\n";
}
int main()
{
        TellTime time1, time2;
        cout << "Enter Data for 2nd Object\n";
        time2.InputData();
        cout << "Data for 1st Object: \n";
        time1.PrintData();
        cout << "Data for 2nd Object: \n";
        time2.PrintData();
        return 0;
}
```



Activities
Terminal
Apr 18 3:33 PM
~/Desktop/cpppp.cpp - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
tree.c
cpppp.cpp
Line 41, Column 2
Tab Size: 4
C++

```
#include <bits/stdc++.h>
using namespace std;
class TellTime
{
public:
    int sec;
    string myname;
    //Default Constructor
    TellTime();
    void InputData();
    void PrintData();
};
TellTime::TellTime()
{
    sec = 100;
    myname = "A7A8";
}
void TellTime::InputData()
{
    cout << "\tEnter time in seconds : ";
    cin >> sec;
    cout << "\tEnter Name: ";
    cin.ignore();
    getline(cin,myname);
}
void TellTime::PrintData()
{
    cout << "\tTime is: " << sec/60 << " Minutes and " << sec%60 << " seconds\n";
    cout << "\tName is: " << myname << "\n";
}
int main()
{
    TellTime time1, time2;
    cout << "Enter Data for 2nd Object\n";
    time2.InputData();
    cout << "Data for 1st Object: \n";
    time1.PrintData();
    cout << "Data for 2nd Object: \n";
    time2.PrintData();
    return 0;
}
```

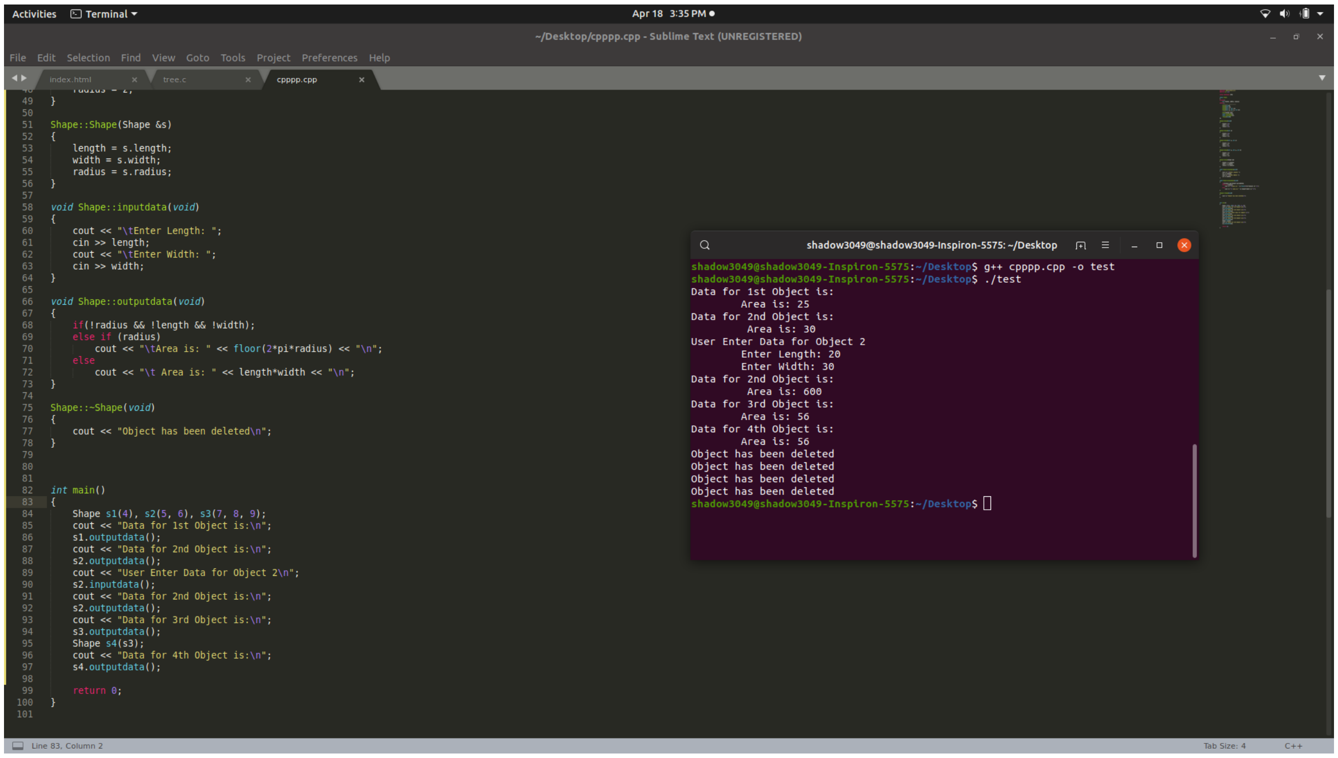
shadow3049@shadow3049-Inspiron-5575: ~/Desktop

```
shadow3049@shadow3049-Inspiron-5575:~/Desktop$ ./test
Enter Data for 2nd Object
        Enter time in seconds : 500
        Enter Name: Shivank Mittal
Data for 1st Object:
        Time is: 1 Minutes and 40 seconds
        Name is: A7A8
Data for 2nd Object:
        Time is: 8 Minutes and 20 seconds
        Name is: Shivank Mittal
shadow3049@shadow3049-Inspiron-5575:~/Desktop$
```

7.

```
#include <bits/stdc++.h>
#define pi 3.14

using namespace std;

class Shape
{
private:
	int length, width, radius;
public:
	//Default Constructor
	Shape(void);
	Shape(int x);
	Shape(int x, int y);
	Shape(int x, int y, int z);
	//Copy Constructor
	Shape(Shape &s);
	void inputdata();
	void outputdata();
	~Shape(void);
};

Shape::Shape(void)
{
	length = 0;
	width = 0;
	radius = 0;
}

Shape::Shape(int x)
{
	length = 0;
	width = 0;
	radius = x;
}

Shape::Shape(int x, int y)
{
	length = x;
	width = y;
	radius = 0;
```

```
}

Shape::Shape(int x, int y, int z)
{
        length = x;
        width = y;
        radius = z;
}

Shape::Shape(Shape &s)
{
        length = s.length;
        width = s.width;
        radius = s.radius;
}

void Shape::inputdata(void)
{
        cout << "\tEnter Length: ";
        cin >> length;
        cout << "\tEnter Width: ";
        cin >> width;
}

void Shape::outputdata(void)
{
        if(!radius && !length && !width);
        else if (radius)
                cout << "\tArea is: " << floor(2*pi*radius) << "\n";
        else
                cout << "\t Area is: " << length*width << "\n";
}

Shape::~Shape(void)
{
        cout << "Object has been deleted\n";
}



int main()
{
        Shape s1(4), s2(5, 6), s3(7, 8, 9);
        cout << "Data for 1st Object is:\n";
        s1.outputdata();
        cout << "Data for 2nd Object is:\n";
        s2.outputdata();
        cout << "User Enter Data for Object 2\n";
        s2.inputdata();
        cout << "Data for 2nd Object is:\n";
        s2.outputdata();
        cout << "Data for 3rd Object is:\n";
        s3.outputdata();
        Shape s4(s3);
        cout << "Data for 4th Object is:\n";
        s4.outputdata();

        return 0;
}
```


Activities
Terminal
Apr 18 3:35 PM
~/Desktop/cpppp.cpp - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
index.html
tree.c
cpppp.cpp
shadow3049@shadow3049-Inspiron-5575: ~/Desktop
Line 83, Column 2
Tab Size: 4
C++

```
    radius = 2;
}

Shape::Shape(Shape &s)
{
    length = s.length;
    width = s.width;
    radius = s.radius;
}

void Shape::inputdata(void)
{
    cout << "\tEnter Length: ";
    cin >> length;
    cout << "\tEnter Width: ";
    cin >> width;
}

void Shape::outputdata(void)
{
    if(!radius && !length && !width);
    else if (radius)
        cout << "\tArea is: " << floor(2*pi*radius) << "\n";
    else
        cout << "\t Area is: " << length*width << "\n";
}

Shape::~Shape(void)
{
    cout << "Object has been deleted\n";
}



int main()
{
    Shape s1(4), s2(5, 6), s3(7, 8, 9);
    cout << "Data for 1st Object is:\n";
    s1.outputdata();
    cout << "Data for 2nd Object is:\n";
    s2.outputdata();
    cout << "User Enter Data for Object 2\n";
    s2.inputdata();
    cout << "Data for 2nd Object is:\n";
    s2.outputdata();
    cout << "Data for 3rd Object is:\n";
    s3.outputdata();
    Shape s4(s3);
    cout << "Data for 4th Object is:\n";
    s4.outputdata();

    return 0;
}
```

```
shadow3049@shadow3049-Inspiron-5575:~/Desktop$ g++ cpppp.cpp -o test
shadow3049@shadow3049-Inspiron-5575:~/Desktop$ ./test
Data for 1st Object is:
        Area is: 25
Data for 2nd Object is:
         Area is: 30
User Enter Data for Object 2
        Enter Length: 20
        Enter Width: 30
Data for 2nd Object is:
         Area is: 600
Data for 3rd Object is:
        Area is: 56
Data for 4th Object is:
        Area is: 56
Object has been deleted
Object has been deleted
Object has been deleted
Object has been deleted
shadow3049@shadow3049-Inspiron-5575:~/Desktop$ 
```