4

IT 2022: Object Oriented Programming

Data Abstraction

Sandali Goonatilleke

Department of Computer Engineering

Module Content

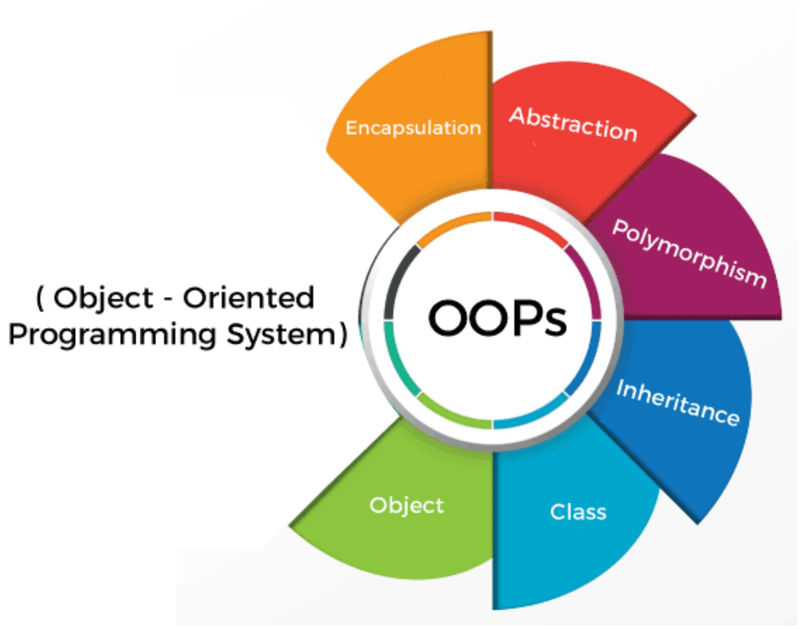
- Introduction to Object-Oriented Programming in C++
- Classes & Objects
- Constructors & Destructors
- Data Abstraction
- Encapsulation
- Composition
- Inheritance
- Polymorphism

Overview

- What is Abstraction
- Real life example of Abstraction in C++
- Advantages of Data Abstraction
- Abstraction using Classes, Abstraction in Header files and Abstraction using access specifiers

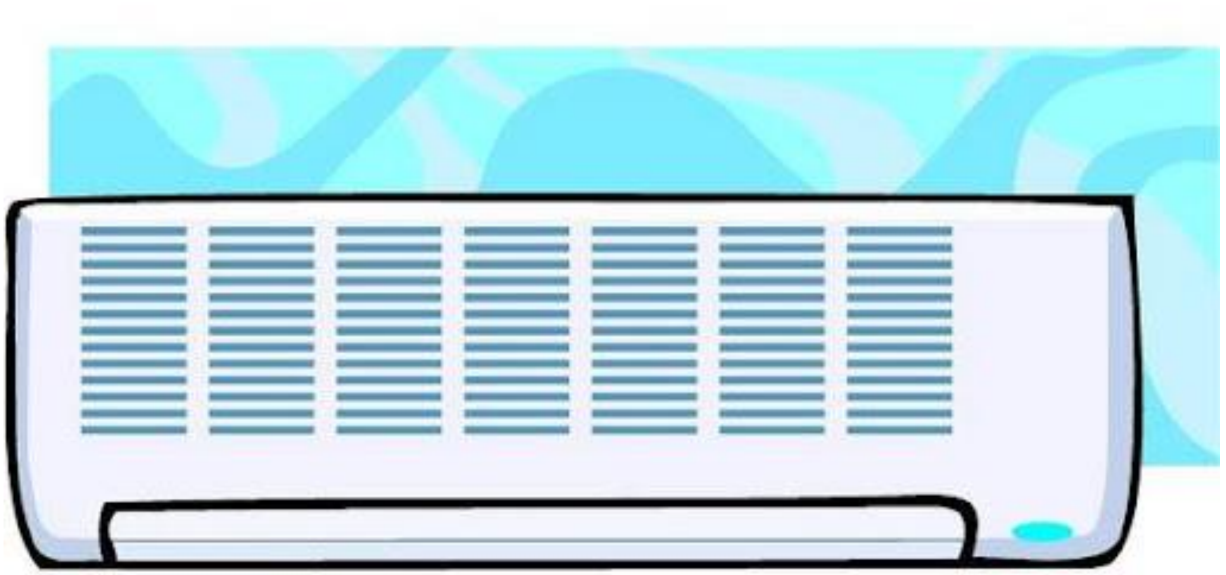
Introduction

- OOP offers various features to write programs with various concepts that help to minimize problems and increase flexibility in the program
- One of the features of object-oriented programming is Data abstraction



Abstraction – Real Life Example 1

Air Conditioner



Abstraction – Real Life Example 1

Air Conditioner

What are the tasks user can perform?

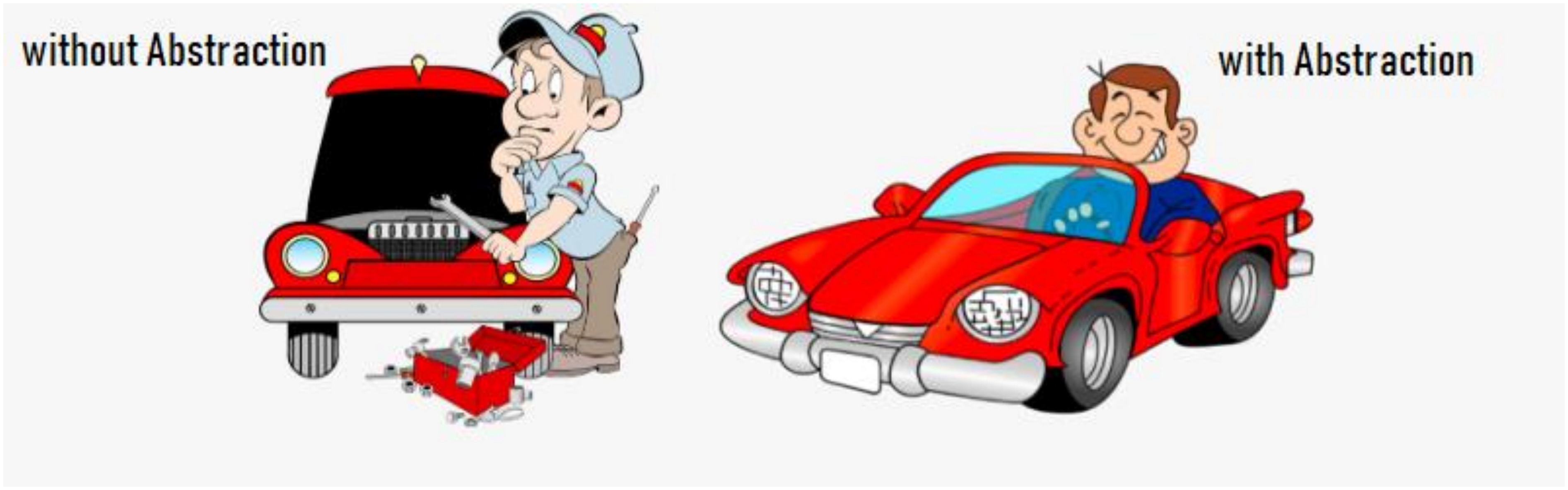
- Can turned ON or OFF the machine
- Can change the temperature
- Can change the mode, and other external components such as fan, swing

Abstraction

What is the problem?

- User don't know the internal details of the AC or the way how it works internally
- Thus, we can say that AC seperates the implementation details from the external interface

Abstraction – Real Life Example 2



Abstraction – Real Life Example 3

dreamstime

Abstraction in C++

- Abstraction refers to the act of representing the crucial requisite features without including the additional explanations of such features
- It means providing the end-user with their needs but without the details of how the needs were fulfilled

- Types of Abstraction in C++

  -Data Abstraction

  -Control Abstraction

Abstraction in C++

Data Abstraction: It hides the information about the data

Control Abstraction: It hides the information about the implementation

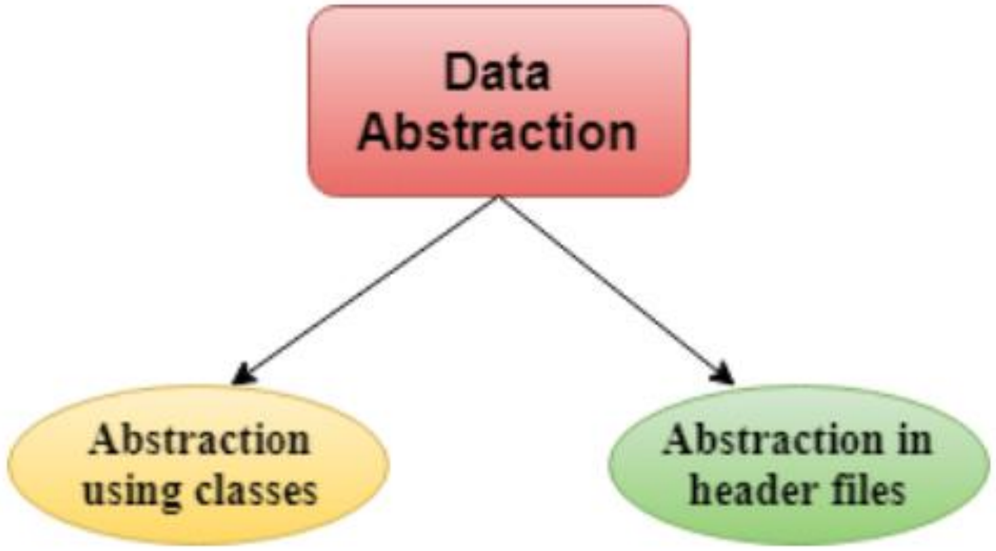
Data Abstraction

- It is a process of providing only the essential details to the outside world and hiding the internal details
- Representing only the essential details in the program
- Data Abstraction is a programming technique that depends on the seperation of the interface and implementation details of the program

Data Abstraction

Data Abstraction can be achieved in two ways;

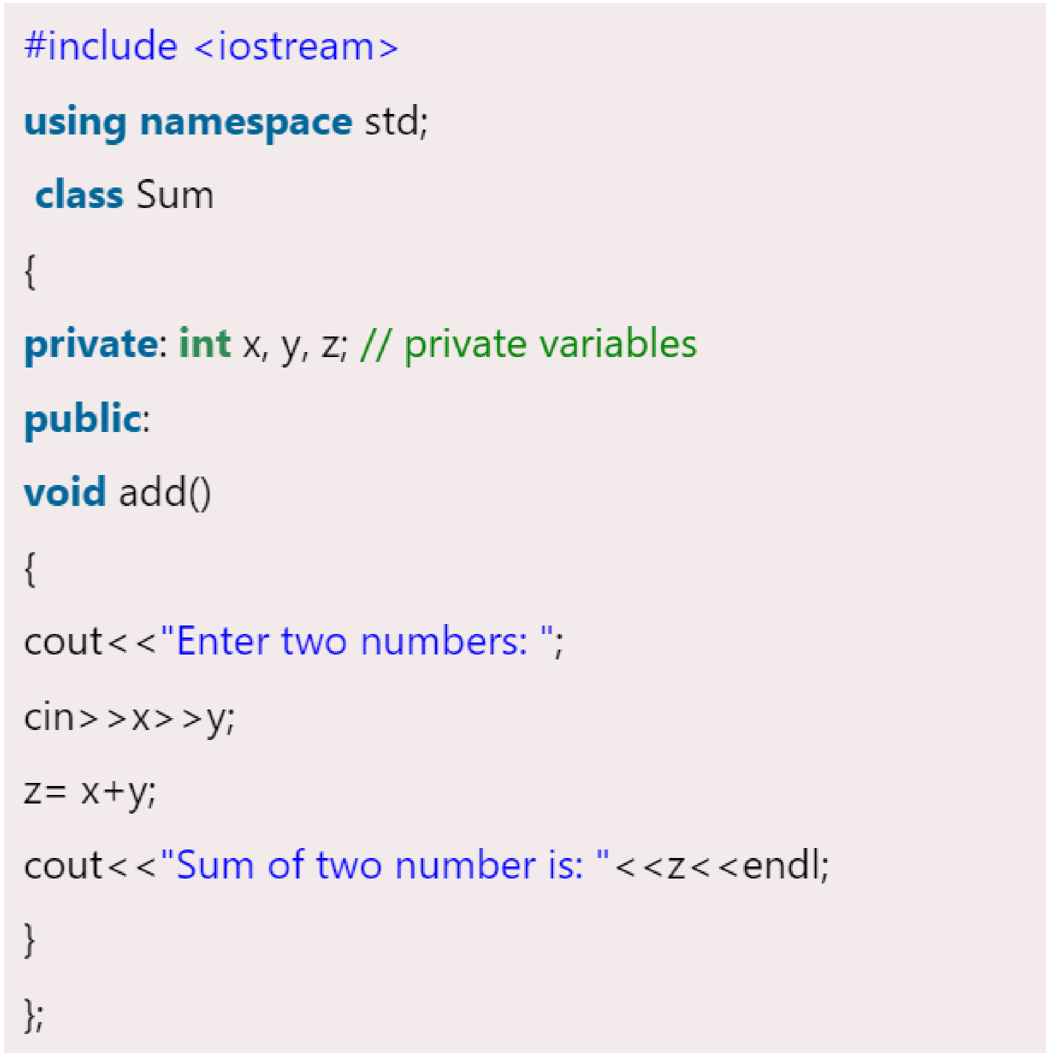
- Abstraction using classes
- Abstraction in header files



Abstraction using Classes

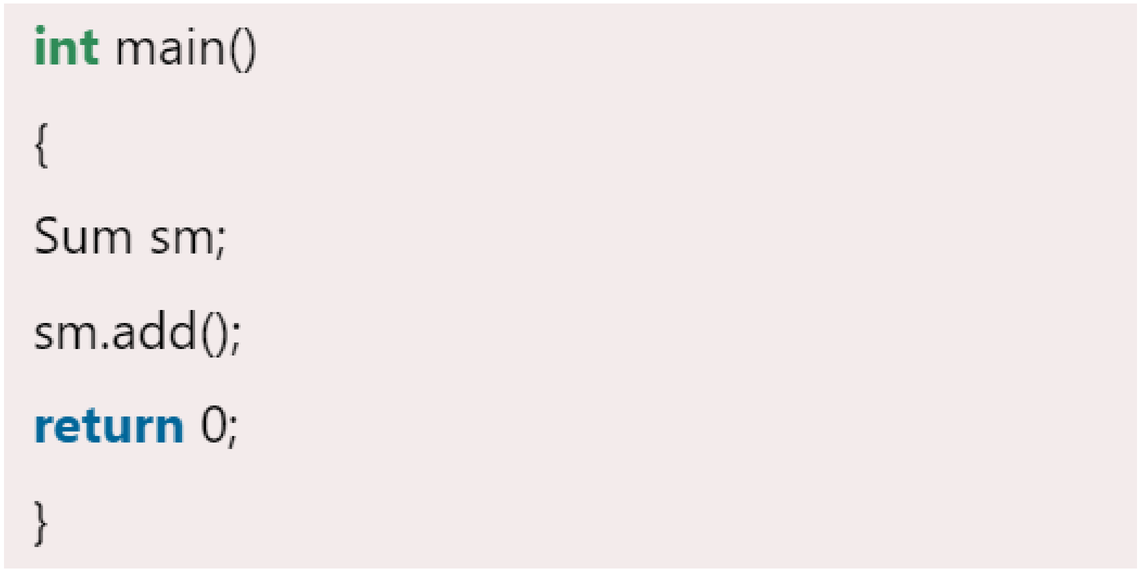
- An abstraction can be achieved using classes
- A class is used to group all the data members and member functions into a single unit by using the access specifiers
- A class has the responsibility to determine which data member is to be visible outside and which is not

Abstraction using Classes - Example



```
#include <iostream>
using namespace std;
 class Sum
{
private: int x, y, z; // private variables
public:
void add()
{
cout<<"Enter two numbers: ";
cin>>x>>y;
z= x+y;
cout<<"Sum of two number is: "<<z<<endl;
}
};
```

Abstraction using Classes - Example



```
int main()
{
Sum sm;
sm.add();
return 0;
}
```

Abstraction in Header Files

- An another type of abstraction is header file
- For example, **pow()** function available is used to calculate the power of a number without actually knowing which algorithm function uses to calculate the power
- Thus, we can say that header files hides all the implementation details from the user

Abstraction in Header Files - Example

```
#include <iostream>

#include<math.h>

using namespace std;

int main()

{

 int n = 4;

   int power = 3;

   int result = pow(n,power);         // pow(n,power) is the  power function

   std::cout << "Cube of n is : " <<result<< std::endl;

   return 0;

}
```

Access Specifiers Implement Abstraction

- **Public specifier:** When the members are declared as public, members can be accessed anywhere from the program
- **Private specifier:** When the members are declared as private, members can only be accessed only by the member functions of the class

Advantages of Data Abstraction

- Class internals get protected from inadvertent user-level errors
- Programmer does not have to write the low-level code
- Code duplication is avoided and so programmer does not have to go over again and again fairly common tasks every time to perform similar operation
- The main idea of abstraction is code reuse and proper partitioning across classes

Advantages of Data Abstraction

- For small projects, this may not seem useful but for large projects, it provides conformity and structure as it provides documentation through the abstract class contract
- It allows internal implementation details to be changed without affecting the users of the abstraction

Thank You!