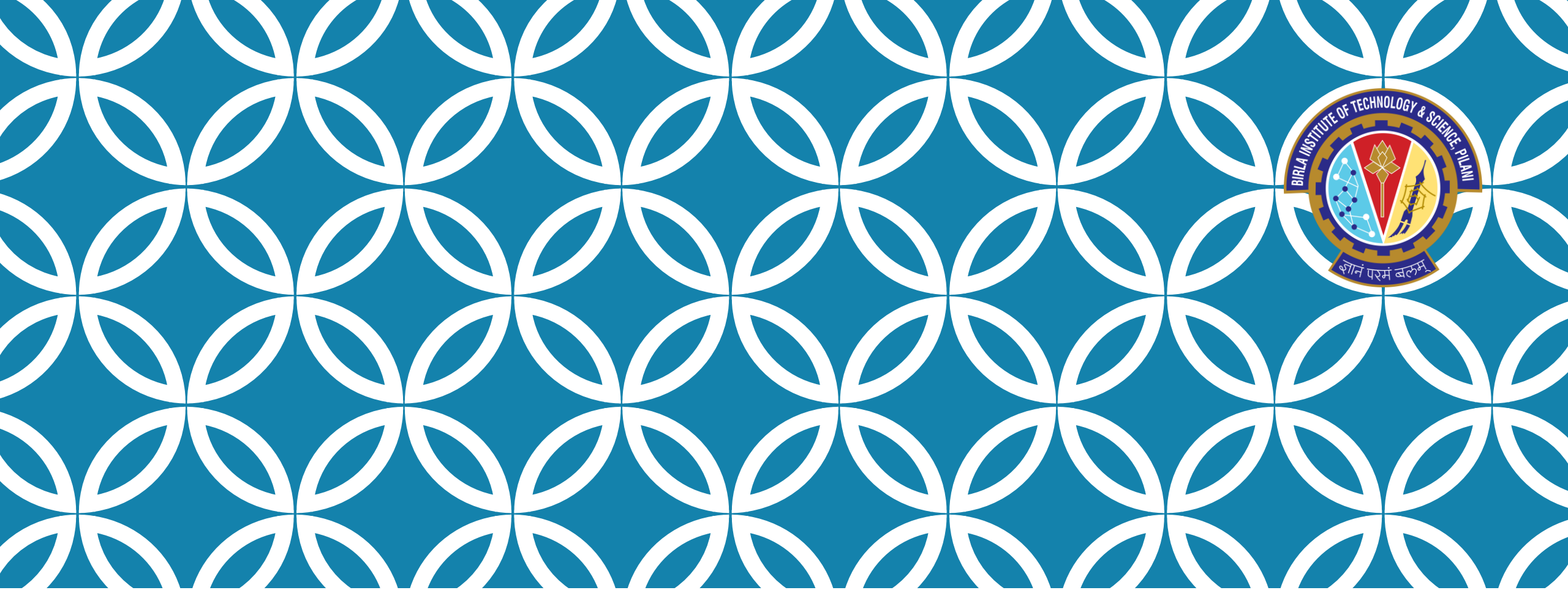


BITS F232: FOUNDATIONS OF DATA STRUCTURES & ALGORITHMS (1ST SEMESTER 2023-24) INTRODUCTION TO C++

Chittaranjan Hota, PhD
Sr. Professor of Computer Sc.
BITS-Pilani Hyderabad Campus
hota[AT]hyderabad.bits-pilani.ac.in

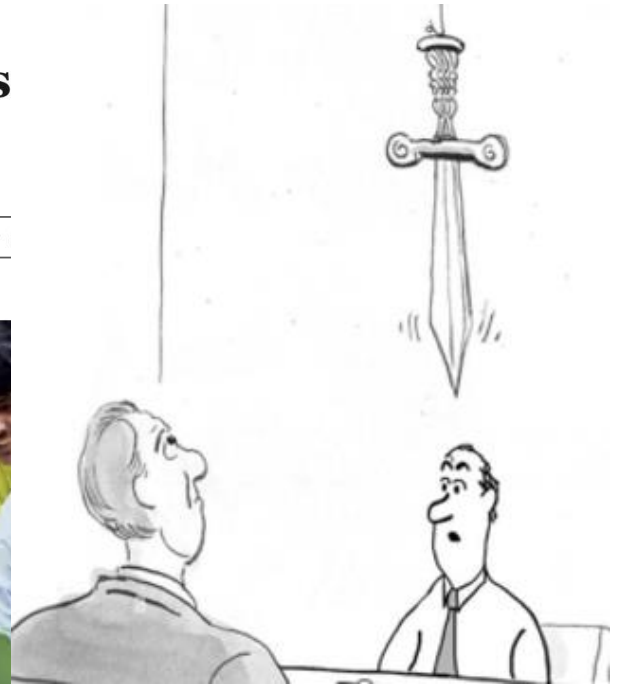
COURSE CONTENT AND ADMINISTRATION

Evaluations:

Component	Duration	Weightage	Date & Time	Nature
Mid sem Test	90 mins.	20%	13/10/2023 (4:00 pm)	Closed Book
Class Interaction	In the class	15%	In the class (best 25/30)	Quiz (Open)
Lab Interaction	In the lab	15%	In the lab (best 10/ 13)	Quiz (Open)
Lab Test (One)	60 mins.	10%	To be announced	Open Book
Programming Assignments (1)	-	10%	To be announced	Take Home
Comprehensive examination	180 mins.	30%	19/12/2023, AN	Part Open

ates

s



Sword of Damocles

Course Content:

Intro to C++, Elementary data structures and algo. analysis techniques, More common data structures, Advanced data structures, Understanding algorithmic techniques.

Course notices and material: google class page

Chamber consultation hour: Every Monday (5 to 6 pm)

WHY C++ FOR BITS F232?

- ✓Developed (in 1979) by Bjarne Stroustrup: Why is it called C++?
- ✓Mid-level: Used for both application level and system level programming tasks.
- ✓Has Object-oriented features improving the quality and reusability of the program.
- ✓Rich library (iostream, iomanip, cmath, cstdlib, iterator, algorithm etc.), Efficiency and speed (competitive coding) …

- ✓Adobe (Photoshop, Illustrator etc are developed using C++, Microsoft used C++ for all of its versions of OS starting from Windows 95, Microsoft Office too is developed using C++, Apple uses C++ to code its OS, MySQL also is written using C++, Mozilla uses a subset of C++, Amazon AWS SDK for C++. Meta, Capgemini, IBM, …

C++ EXAMPLES

```
#include <iostream>

int main( )
{
  int x , y;
  std::cout << "Please enter two numbers: ";
  std::cin >> x >> y;
  int sum = x + y;
  std::cout << "Sum = " << sum << std::endl;
  return EXIT_SUCCESS;
}
```

```
Please enter two numbers: 45 7
Sum = 52


...Program finished with exit code 0
Press ENTER to exit console.
```

```
#include <iostream>
using namespace std;
bool  testSum (int  a[ ], int  n) {
   int  sum = 0;
   for (int  i = 0; i < n; i++)
      sum +=  a[ i ];
   return (sum % 2 ) == 0;
}
int  main( )
{
   int a [ 6 ] = {4, 4, 7, 6, 5, 2};
   bool  result = testSum ( a, 6);
   if (result)
      cout << "Sum of all the nos. is even\n";
   else
      cout << " Sum of all the nos. is odd\n";
return  EXIT_SUCCESS;
}
```

```
Sum of all the nos. is even
```

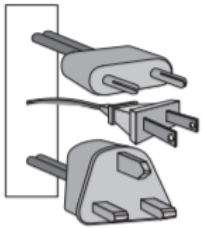
OBJECT-ORIENTED DESIGN: GOALS AND PRINCIPLES

What is Object-Oriented Design?

- Style of writing computer programs using objects, and their interactions. (Minor degree admissions at BITS, Hyderabad: How many objects and what are their interactions)

What are the Design Goals?

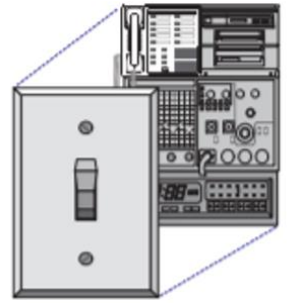
- Robustness
- Adaptability
- Reusability



(Design Principles)

Abstraction (ADTs are realized by classes in C++)

Encapsulation (Access to data is provided through member functions)



Modularity (different components): supported through hierarchy.

CLASSES IN C++

- Class: A user-defined type or data structure that has data and functions as its members whose access is governed by the access specifiers.
- Object: A variable declared to be of some class, hence includes both data and functions for that object.
- Usage: A variable is an instance of a type. Similarly, an object is an instance of a class.
- Ex:

```
class  Passerger {
    private:
        string      name;
        MealType    mealPref;
        bool        isFreqflyer;
        string      freqFlyerNo;
    public:
        Passenger( );
        bool  isFrequentFlyer( ) const { return  isFreqFlyer; }
        void makeFrequentFlyer(const  string&  newFreqFlyerNo) {
              isFreqFlyer = true;
              freqFlyerNo = newFreqFlyerNo;
              }
} ;
```

Member variables

Member functions

```
Passenger pass;
if (!pass.isFrequentFlyer()) { pass.makeFrequentFlyer ("12345"); }
```

ILLEGAL: pass.name = "Amit";

ACCESS MODIFIERS: PUBLIC, PRIVATE AND PROTECTED

main.cpp

```cpp
#include<iostream>
using namespace std;

class Circle
{
    public:
        double radius;

        double  compute_area()
        {
            return 3.14*radius*radius;
        }

};

int main()
{
    Circle obj;

    obj.radius = 7.2;

    cout << "Radius is: " << obj.radius << "\n";
    cout << "Area is: " << obj.compute_area();
    return 0;
}
```

```
Radius is: 7.2
Area is: 162.778

...Program finished with exit code 0
Press ENTER to exit console.
```

main.cpp

```cpp
#include<iostream>
using namespace std;

class Circle
{
    private:
        double radius;

    public:
        double  compute_area()
        {
            return 3.14*radius*radius;
        }

};

int main()
{
    Circle obj;

    obj.radius = 7.2;

    cout << "Area is:" << obj.compute_area();
    return 0;
}
```

input

Compilation failed due to following error(s).

```
main.cpp:21:9: error: 'double Circle::radius' is private within this context
     obj.radius = 7.2;
         ^~~~~~
main.cpp:7:16: note: declared private here
         double radius;
                ^~~~~~
```

main.cpp

```cpp
using namespace std;

class Circle
{
    private:
        double radius;
    public:
        void compute_area(double r)
        {
            radius = r;

            double area = 3.14*radius*radius;

            cout << "Radius is: " << radius << endl;
            cout << "Area is: " << area;
        }

};

int main()
{
    Circle obj;

    obj.compute_area(7.2);


    return 0;
}
```

```
Radius is: 7.2
Area is: 162.778

...Program finished with exit code 0
Press ENTER to exit console.
```

CONTINUED...

main.cpp

```cpp
#include<iostream>
using namespace std;
// base class
class Parent
{
    // protected data members
    protected:
        int id_protected;
};

// sub class or derived class from public base class
class Child : public Parent
{
    public:
    void setId(int id)
    {
        // Child class is able to access the inherited
        // protected data members of base class

        id_protected = id;

    }

    void displayId()
    {
        cout << "id_protected is: " << id_protected << endl;
    }
};
```

```cpp

// main function
int main() {

    Child obj1;

    // member function of the derived class can
    // access the protected data members of the base class

    obj1.setId(12345);
    obj1.displayId();
    return 0;
}
```

```
id_protected is: 12345


...Program finished with exit code 0
Press ENTER to exit console.
```

CONSTRUCTORS IN C++

```
1  #include <iostream>
2  using namespace std;
3
4  class Rectangle {
5  public:
6      void set_values(int x,int y) {width=x; height=y;}
7      int area() {return width*height;}
8  private:
9      int width, height;
10 };
11
12 int main () {
13   Rectangle rect;
14   rect.set_values (7,9);
15   cout << "area of rectangle: " << rect.area();
16   return 0;
17 }
```

g++ 17 GCC 9.1.0

CommandLine Arguments

Result

CPU Time: 0.00 sec(s), Memory: 3272 kilobyte(s)

```
area of rectangle: 63
```

```
14   cout << "area of rectangle: " << rect.area();
15   rect.set_values (7,9);
```

```
area of rectangle: -897836400
```

THANK YOU!

Next Class: C++ to be continued…