First Year First Semester Course

M.Tech. (CS) [Batch 2021-23]

Lecture #11

# Introduction to Programming

Principles of Object-Oriented Programming


**Course Instructor:**

**Dr. Monidipa Das**

**DST-INSPIRE Faculty**

**Machine Intelligence Unit (MIU), Centre for Artificial Intelligence and Machine Learning (CAIML)**

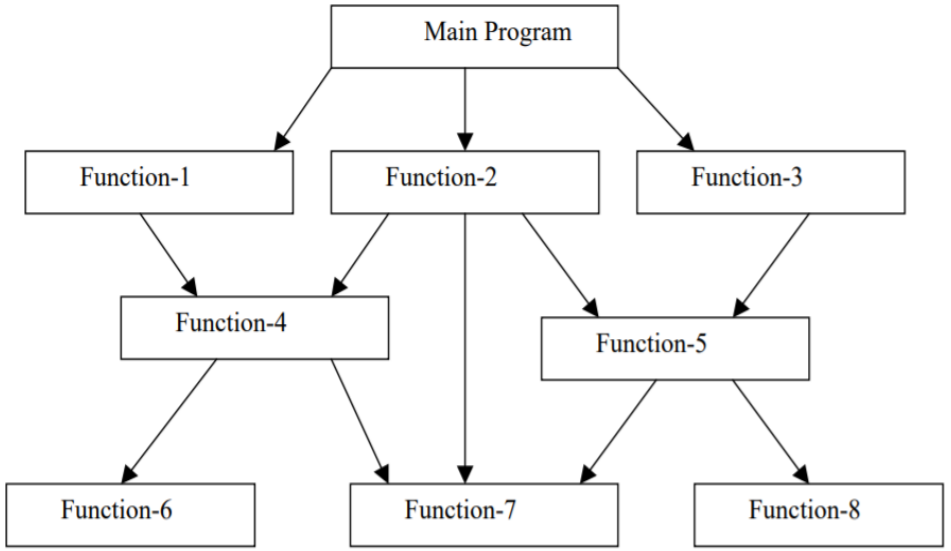
**Indian Statistical Institute (ISI) Kolkata, India**

# Procedure-Oriented vs. Object-Oriented



- Procedure-Oriented Programming

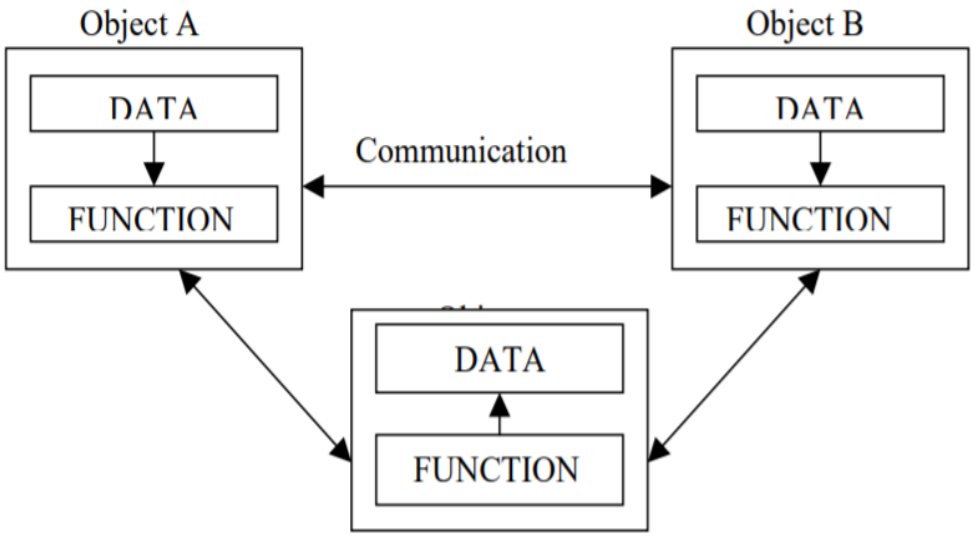
• Emphasis is on doing things (algorithms).
• Large programs are divided into smaller programs known as functions.
• Most of the functions share global data.
• Data move openly around the system from function to function.
• Functions transform data from one form to another.
• Employs top-down approach in program design.



# Procedure-Oriented vs. Object-Oriented



- Object Oriented Programming (OOP)




- Emphasis is on data rather than procedure.
- Programs are divided into what are known as objects.
- Data is hidden and cannot be accessed by external function.
- Objects may communicate with each other through function.
- New data and functions can be easily added whenever necessary.
- Follows bottom up approach in program design.

# Basic Concepts of OOP

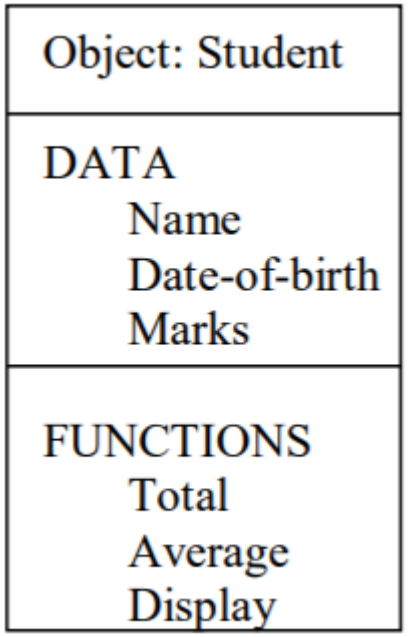


- Objects
- Classes
- Data abstraction and encapsulation
- Inheritance
- Polymorphism
- Dynamic binding
- Message passing

# Objects



- The fundamental idea behind object oriented approach is to combine both data and function into a single unit and these units are called objects
- Basic run-time entities in an object-oriented system.
- May represent a person, a place, a bank account, a table of data or any item.



# Classes



- Collection of objects of similar type.

- In C ++ a class is a new data type that contains member variables and member functions that operate on the variables.

```
fruit apple;
```

# Data abstraction and encapsulation



- Abstraction
  - Act of representing essential features without including the background details or explanations.
  - Classes use the concept of abstraction and are defined as a list of abstract attributes size, cost, etc., and functions to operate on the attributes.

- Encapsulation
  - Wrapping up of data and function into a single unit (called class).
  - The data is not accessible to the outside world and only those functions which are wrapped in the class can access it.

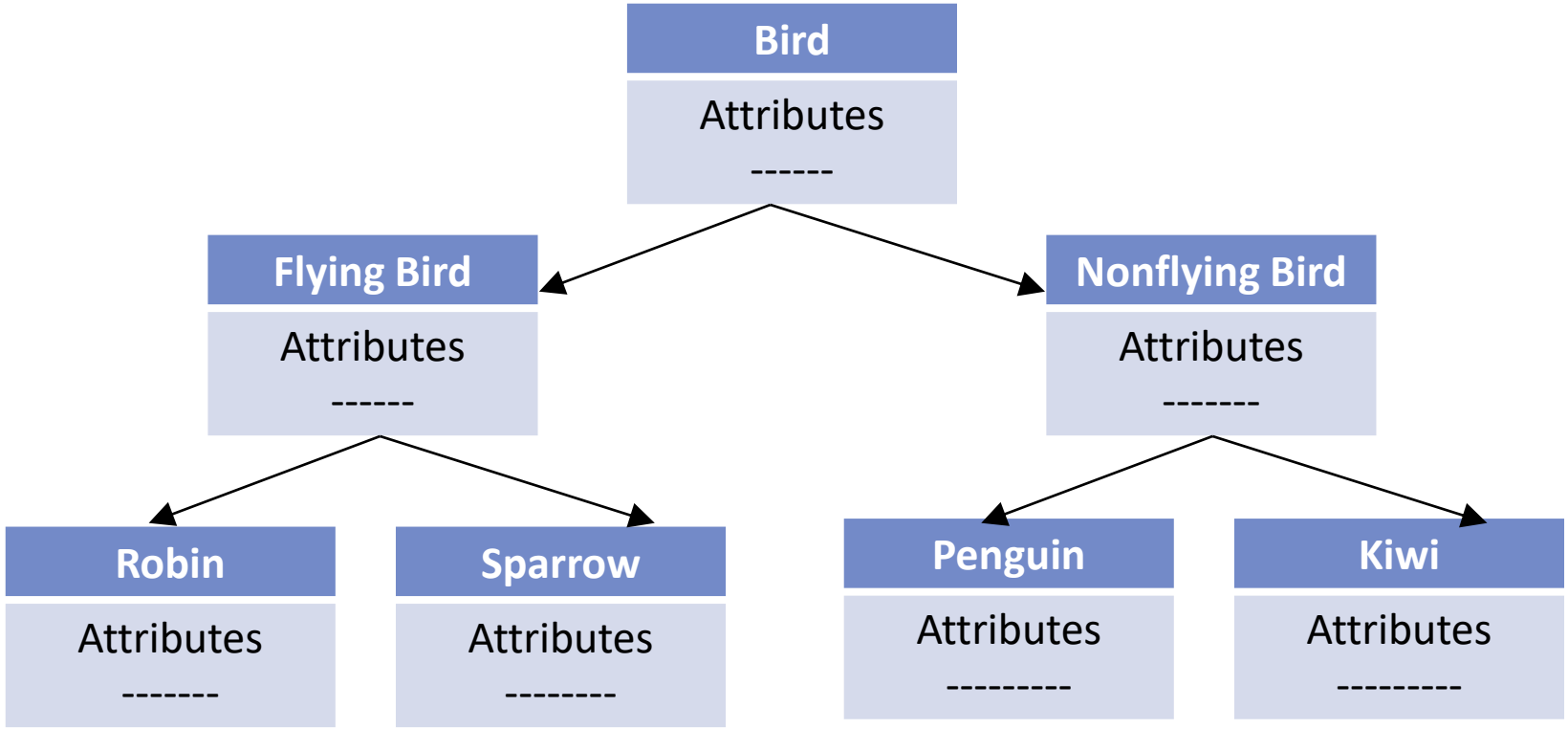
# Inheritance



- The process by which objects of one class acquire the properties of another class.
- In the concept of inheritance provides the idea of re-usablity.
- This is possible by deriving a new class from an existing class. The new class will have the combined features of both the classes.

# Inheritance





# Polymorphism



- Polymorphism means the ability to take more than one form.
- A language feature that allows a function or operator to be given more than one definition.
- The types of the arguments with which the function or operator is called determines which definition will be used.
- The process of making an operator to exhibit different behaviors in different instances in called *operator overloading*
- Using a single function name to perform different types of tasks is known as *function overloading*

# Dynamic binding



- Binding refers to the linking of a procedure call to the code to the executed in response to the call.

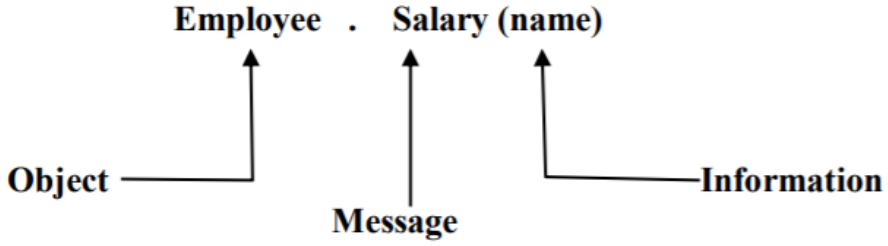
- Dynamic binding means the code associated with a given procedure call is not known until the time of the call at run-time.

- It is associated with a polymorphic reference
- Depends upon the dynamic type of that reference.

# Message passing



- An object oriented program consists of a set of objects that communicate with each other.
- A message for an object is a request for execution of a procedure and therefore will invoke a function (procedure) in the receiving object that generates the desired result.
- Message passing involves specifying the name of the object, the name of the function (message) and information to be sent.



# Benefits of OOP



1. Through inheritance we can eliminate redundant code and extend the use of existing classes.

2. We can build programs from the standard working modules that communicate with one another, rather than having to start writing the code from scratch. This leads to saving of development time and higher productivity.

3. This principle of data hiding helps the programmer to build secure programs that can't be invaded by code in other parts of the program.

4. It is possible to have multiple instances of an object to co-exist with out any interference.

5. It is easy to partition the work in a project based on objects.

6. Object-oriented systems can be easily upgraded from small to large systems.

7. Message passing techniques for communication between objects makes the interface description with external systems much simpler.

8. Software complexity can be easily managed.

# Basics of C++



- Object oriented programming language
- Developed by Jarney Stroustrup at AT & T Bell lab, USA
- Was previously called ‘C with classes’.
- Additional facilities offered: classes, inheritance, function overloading, operator overloading

# A Simple C++ Program



```
#include <iostream>  //include header file
using namespace std;
int main ()
{
   cout << "Hello World!";  //C++ statement
   return 0;
}
```

# Program features



- C++ Comments:
  - C++ introduces a new comment symbol //(double slash).
  - Multi line comments can be written as follows:

```
// this is an example of
// c++ program
// thank you
```

  - The c comment symbols /* ….*/ are still valid and more suitable for multi line comments.

# Program features



- Output Operator:
  - The statement *cout << "Hello World!";* causes the string within quotation marks to be displayed on the screen
  - Also called the ‘insertion’ or ‘put to’ operator
  - Directs the information on its right to the object on its left

# Program features



- Input Operator:
  - The statement *cin>> number 1;* is an input statement and causes the program to wait for the user to type in a number.
  - The operator >> is known as *get from* operator.
  - It extracts value from the keyboard and assigns it to the variable on its right

# Program features



- Cascading Of I/O Operator:

```
cout<<“sum”<<sum<<“\n”;
cout<<“sum”<<sum<<“\n”<<“average”<<average<<“\n”;
cin>>number1>>number2;
```

- Return Statement:
  - Every main ( ) in C++ should end with a return (0) statement, otherwise a warning or an error might occur.

# Program features



- Namespace:
  - Defines a scope for the identifiers that would be used in a program

```
using namespace std;
```

  - std is the namespace where ANSI C++ standard class libraries are defined.

  - using and namespace are the new keywords of C++

# C++ TOKENS



- The smallest individual units in program are known as tokens. C++ has the following tokens.
  i. Keywords
  ii. Identifiers
  iii. Constants
  iv. Strings
  v. Operators

# C++ KEYWORDS



| | | | |
|---|---|---|---|
| asm | double | new | switch |
| auto | else | operator | template |
| break | enum | private | this |
| case | extern | protected | throw |
| catch | float | public | try |
| char | for | register | typedef |
| class | friend | return | union |
| const | goto | short | unsigned |
| continue | if | signed | virtual |
| default | inline | sizeof | void |
| delete | long | struct | while |

# C++ IDENTIFIERS



- Identifiers refers to the name of variable , functions, array, class etc. created by programmer.

1. Only alphabetic chars, digits and under score are permitted.
2. The name can’t start with a digit.
3. Upper case and lower case letters are distinct.
4. A declared keyword can’t be used as a variable name.

# Questions?