

CSC 226: OBJECT-ORIENTED PROGRAMMING I C++

(3 UNITS) - COMPULSORY

Introduction to C: Structured programming elements, Structured design principles, abstraction modularity, Stepwise refinement, Structured design techniques. Teaching of a structured programming such as C. Basic Object-Oriented Programming Concepts, Objects, Inheritance, polymorphism, data Abstraction, tools for developing, Compiling, Interrupting and debugging an Object-Oriented programming language. Laboratory exercises in an Object-Oriented Programming Language. C++.

: (0*I) \ ((+main() +p2 +d) = IX

: (0*C) \ ((+main() +p2 - d) = CX

: bns >> " + main() bnd los + doo" >> tuo)

: bns >> IX >> " = IX" >> tuo)

: bns >> CX >> " = CX" >> tuo)

{

} (0 = main() ji 212

: bns >> " sm02 bnd los + doo" >> tuo)

: (0*S) \ d = IX

: bns >> IX >> " = IX" >> tuo)

{

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

++ ID C++ Roots of a Quadratic Equation.

#include <iostream>

#include <cmath>

Using namespace std;

int main() {

float a, b, c, x1, x2, discriminant, realPart, imaginaryPart;

Cout << "Enter Coefficients a, b and c:";

Cin >> a >> b >> c;

discriminant = b * b - 4 * a * c;

if (discriminant > 0) {

x1 = (-b + sqrt(discriminant)) / (2 * a);

x2 = (-b - sqrt(discriminant)) / (2 * a);

Cout << "Roots are real and different." << endl;

Cout << "x1=" << x1 << endl;

Cout << "x2=" << x2 << endl;

}

else if (discriminant == 0) {

Cout << "Roots are real and same." << endl;

x1 = -b / (2 * a);

Cout << "x1=x2=" << x1 << endl;

}

```

else {
    realPart = -b / (2 * a);
    ImaginaryPart = Sqrt(-discriminant) / (2 * a);
    cout << "Roots are Complex and different." << endl;
    cout << "x1=" << realPart << "+" << ImaginaryPart << "i" << endl;
    cout << "x2=" << realPart << "-" << ImaginaryPart << "i" << endl;
}
return 0;
}

```

flowchart
of
Quadratic
Equation

Start

Input a, b, c

$D \leftarrow \sqrt{b \times b - 4 \times a \times c}$

$x_1 \leftarrow (-b + d) / (2 \times a)$

$x_2 \leftarrow (-b - d) / (2 \times a)$

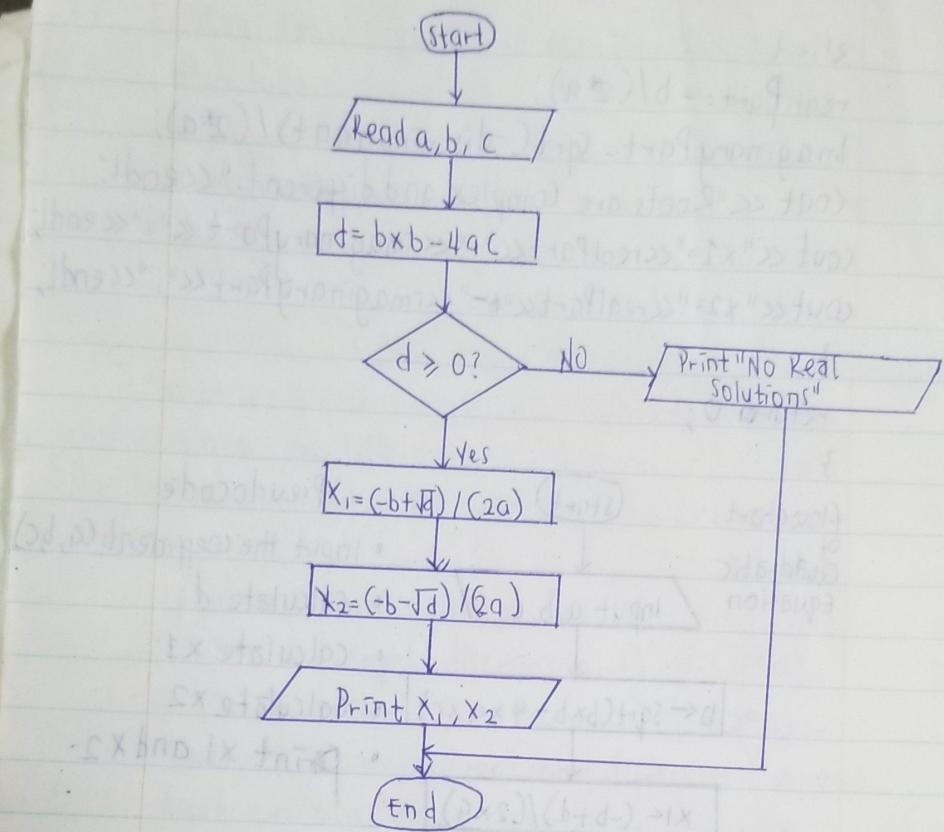
Print x_1, x_2

Stop.

Pseudocode

- Input the coefficients (a, b, c)
- calculate d
- calculate x_1
- calculate x_2
- print x_1 and x_2

Another flowchart of Quadratic equation
Having a decision statement.



Object-Oriented Programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic.

Topic: Intro to OOP.

Objectives.

- Software Crisis, -Software Evaluation
- POP (Procedure Oriented Programming).
- OOP (Object Oriented Programming)
- Basic Concepts of OOP.

Software Crisis

Development in software technology continue to be dynamic, New tools and techniques are announced in Quick Succession.

This has forced the Software Engineers and Industries to continuously look for New approaches to Software design and development and they are becoming more and more Critical in view of the increasing complexity of Software

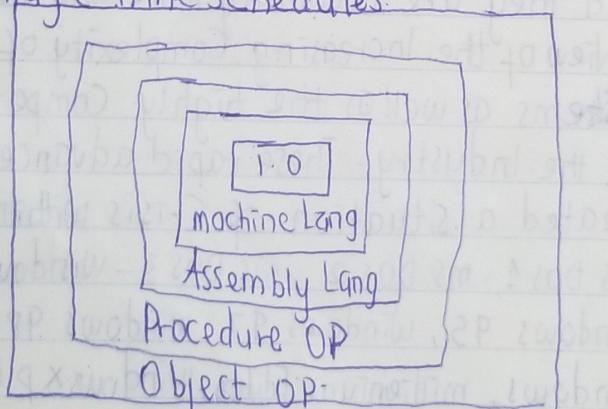
Systems as well as the highly competitive nature of the Industry. These rapid advance appear to have created a situation of Crisis within the Industry.

MS DOS 1, MS DOS 2, MS DOS 3 - Windows 3.11, Windows 95, Windows 97, Windows 98, Windows 2000, Windows Millennium Edition, Windows XP (1, 2, 3), Windows Vista, Windows 7, Windows 8, Windows 10

~~Concept~~ Python is a Compiled interpreted programming language

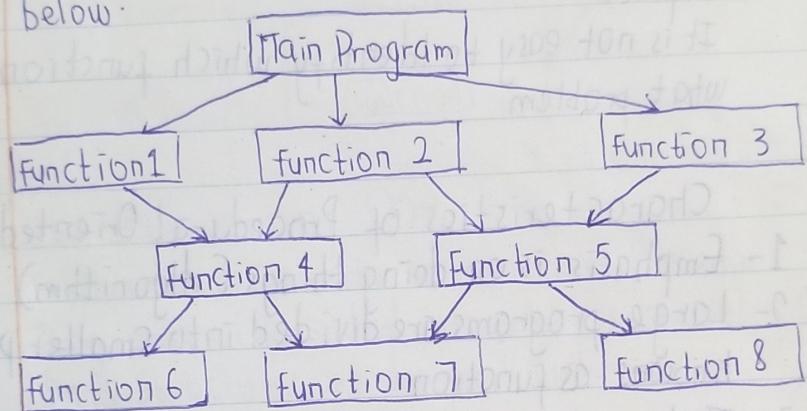
Concept of Space Complexity And Time Complexity.
the following issue needs to be addressed to face the Crisis

- 1 How to Represent Real life Entities of problem in System design.
- 2 Design System with Open Interfaces
- 3 Ensure Reusability and Extensibility of modules
- 4 Develop modules that Are tolerance to Any changes in future
- 5 Improve Software Productivity and decrease Software cost
- 6 Improve the Quality of Software
- 7 Manage Time Schedules



Procedure Oriented Programming

In Procedure Oriented Approach the problem is viewed as the sequence of things to be done such as reading, calculating and printing. A typical structure of Procedural Programming is shown below.



It occupies Space

No Room for Inheritance

Typical Structure Of Procedural Programming

Object

Procedural Oriented Programming is a programming language that follows a Step-by-Step approach to break down a task into a collection of variables and routines (or Subroutines) through a sequence of instructions.

Problems Associated with Procedural Programming.

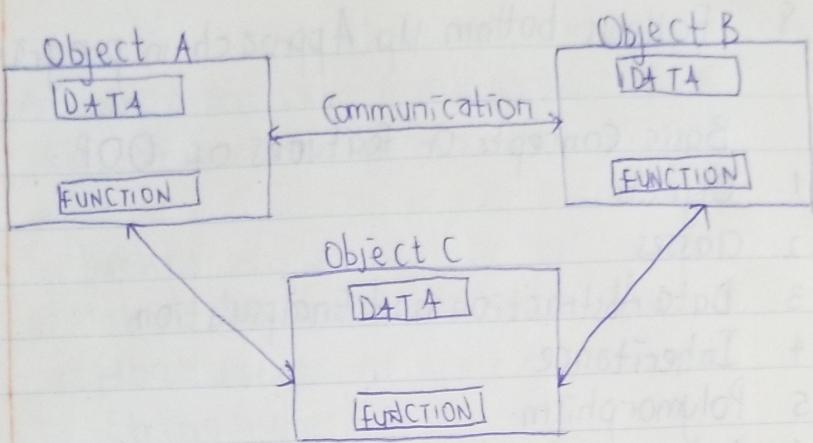
- 1 It is not easy to Model a real life Object.
- 2 Error from access of global Variable because of it is not being well defined
- 3 Error easily Creep into the program because it is not well defined
It is not easy to identify which function causes what problem

Characteristics of Procedural Oriented Programming

- 1- Emphasis is on doing things (Algorithm)
- 2- Large programs are divided into smaller programs known as functions
- 3- Most of the functions Share Global Data.
- 4- Data Moves Openly around the System from function to function
- 5- Functions transform data from One form to another
- 6- Employs top-down Approach in Program Design.

Object Oriented Programming

Organization of data and function in OOP.



Characteristics of OOP:

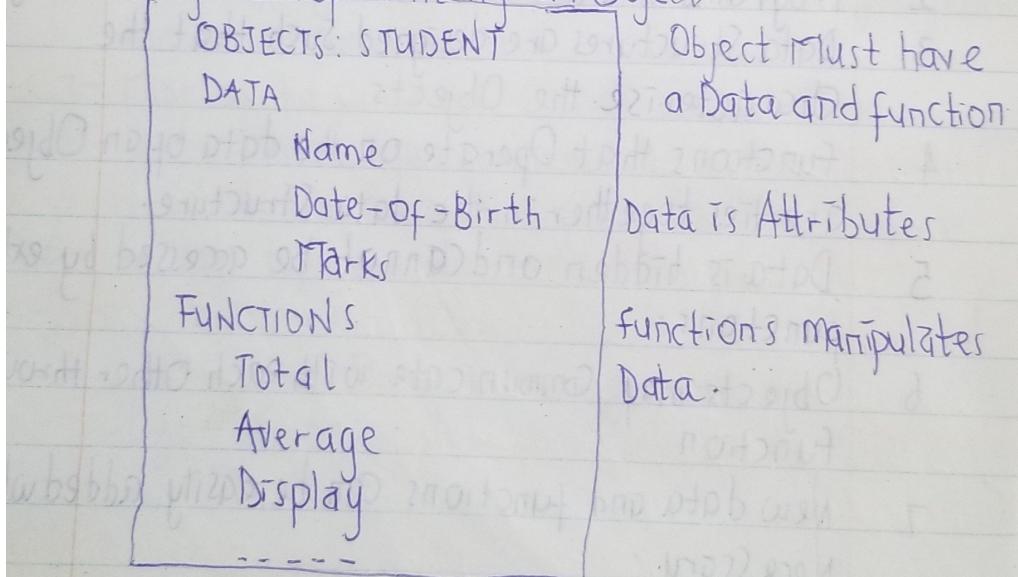
- 1 Emphasis is on data rather than Procedure
- 2 Programs are divided into what is known as Objects
- 3 Data Structures are designed such that the Characterise the Objects
- 4 functions that Operate on the data of an Object are tied together in the data Structure.
- 5 Data is hidden and cannot be accessed by external functions
- 6 Objects may communicate with each other through function
- 7 New data and functions can be easily added when Necessary

8 Follows bottom up Approach in program design.

Basic Concepts or features of OOP.

- 1 Objects
- 2 Classes
- 3 Data Abstraction and Encapsulation
- 4 Inheritance
- 5 Polymorphism
- 6 Dynamic Binding
- 7 Message Passing

Diagram representing an Object.



CLASSES.

An example of C++ class that has three fields only

class Student

{

Public:

Int id; //field or data member

float salary;

String name;

}

Anything after Int Main() { } is Executive.
Differences between -

program Syntax And Program Segment.

Classes And Objects.

Encapsulation.

07-02-2022

The wrapping up of data and function into a single unit called class is known as Encapsulation. These functions provide the interface between the objects data and the program. This insulation of the data from direct access by the program is called data hiding or information hiding. Data Abstraction is the same as data hiding.

Inheritance.

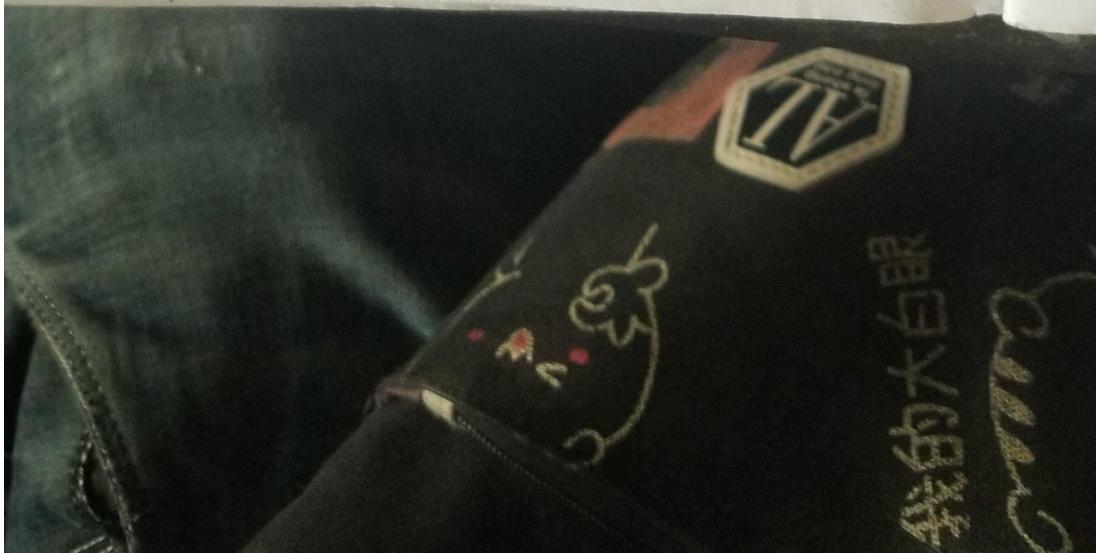
This is the process by which objects of one class acquired the properties of objects of another class. It supports the concept of hierarchical classification.

Polymorphism.

Polymorphism means the ability to take more than one form. Example Shape

Dynamic Binding.

Binding refers to the linking of a procedure call to the code to be executed in response to the particular call. Dynamic Binding means that the code associated with a given procedure call is not known until the time of the



- * call at run-time.
- * Message Passage.
- * Benefits of OOP.

Application of OOP.

The promising Area of Application of OOP includes-

the following:

- 1 Real time Systems.
- 2 Simulation and Modeling.
- 3 Object Oriented databases.
- 4 Hypermedia Hypertext and expertext.
- 5 AI and expert System
- 6 Neural networks and parallel programming.
- 7 Decision Support and Office Automation Systems.
- 8 CIM /CAM /CAD System.

- * Steps Required to Install C++ Compiler.
- * Historical Development of C++
- * Data types of C++

ayglo555@gmail.com

Explain the following terms.

* Objects: Objects are the basic run time entities in an Object-Oriented System. They may represent a person, a place, a bank account, a table of data or any item that the program has to handle. Object is a real world entity, for example, chair, car, pen, mobile, laptop, etc.

In other words, Object is an entity that has state and behavior. Here, State means data and behavior mean functionality. They may also represent User-defined data such as vectors, time and Lists.

* Classes: In C++, class is a group of similar Objects. It is a template from which Objects are created. It can have fields, methods, constructors etc.

Let's see an example of C++ class that has three fields only.

1. class Student
2. {
3. Public:



4. int id; //field or data member
5. float salary; //field or data member
6. String name; //field or data member

A C++ class combines data and Methods for Manipulating the data into one. Classes also determine the forms of Objects. The data and methods contained in a Class are Known as class member.

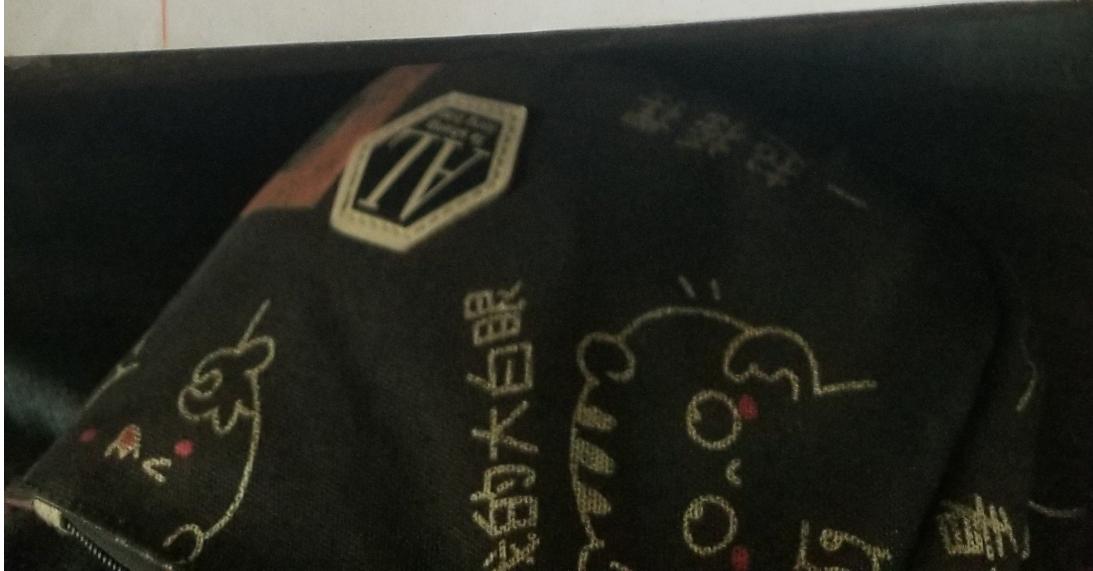
* Data Abstraction:- Data Abstraction is one of the most important features of O-O-P in C++. It means displaying Only essential information and hiding the detail. Data Abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

* Data Encapsulation:- In O.O.P, Data Encapsulation is defined as binding together the data and the functions that manipulate them. Encapsulation also leads to data abstraction or hiding.

Encapsulation in C++

method Variables

class.



* Inheritance - The Capability of a class to derive properties and characteristics from another class ^{for a hierarchy} is called inheritance. Of classes that share a set of attributes and methods.

Sub class - The class that inherits properties from another class. Also known as derived class.

Super class - The class whose properties are inherited by Subclass is called Base class or Super class.

The class whose properties and methods are inherited is known as the Parent class.

* Polymorphism - The word polymorphism means having many form. We can define polymorphism as the ability of a message to be displayed in more than one form. Polymorphism is extensively used in implementing inheritance.

* Dynamic Binding - The code to be executed in response to function call is decided at runtime. C++ has virtual functions to support this.

Message passing:- Objects communicate with one another by sending and receiving information to each other. A message for an Object is a request for execution of a procedure and therefore will invoke a function in the receiving Object that generates the desired results. Message passing involves specifying the name of the Object, the name of the function and information to be sent.

What are the major issues facing the Software Industry today?

- 1 Keeping pace with Innovation.
- 2 Cultural Change.
- 3 Customer Experience
- 4 Data Privacy
- 5 Cybersecurity

What is POP: Procedural Oriented Programming is a programming language that follows a step-by-step approach to break down a task into a collection of variables and routines (or subroutines) through a sequence of instruction.

