Date-08.04.2024

Tender Heart High School, Sector 33B, Chd.

Class: IX Computer Application

Topic: Principles of object Oriented Programming

Teacher: Prabhdeep kaur

Formal language used to communicate with Computer is known as Computer language

Computer languages

- Low Level Languages
- High Level Languages

1. Low Level Languages: are the type of Computer languages in which the Computer recognizes the instructions, without conversion into any other form. They are sort of Cryptic language which are not directly understood by the users.

Low Level Languages

- Machine Language
- Assembly Language

Machine Language is language in which instructions are coaded in terms of binary digits i.e in the form of binary instructions zeros and ones (0's and 1's) and also called machine Code and object Code.

Assembly Language in which the instructions are coaded in terms of mnemonics and op-codes (operation codes), is known as an assembly language.

eg. LD A Enter the value in A 3B

Mnemonics Description op-codes

Date / /

2. High Level Language : To overcome the disadvantages of low level languages, the experts developed another category of languages, which are referred to as the HLL. These languages allow the user to write the instructions in simple English phrases or sentences.
Eg. Basic, C/C++, Java and python.

Compiler and Interpreter

A software that accepts the whole program written in high level language and converts it into its equivalent program in machine language, is known as the compiler.
The program, which the compiler uses for conversion, is known as the source program or source code. The program converted into the machine language is known as the object program or object code.

The software which converts the instructions written in high level languages into their equivalent instructions in machine language, line by line or statement by statement, is known as the Interpreter.

Compiler and Interpreters are basically System softwares, which are also known as the language processors.

Date / /

Tender Heart High School, Sec 33B, Chd

Class IX Computer Application

TYPES OF HIGH LEVEL LANGUAGES

Structure oriented Language	Procedure oriented Language	Object oriented Language

Structure oriented Language: uses modular approach to improve the clarity, quality and the development time of the programming steps. It uses various logical structures like the structure of selective control flow, structure of looping block, structure of subroutines or functions. Eg. ALGOL and PASCAL of SOL

Procedure oriented Programming Language:

This approach allows the users to develop their logic by using a number of functions that would enhance the program's productivity. Eg. BASIC, COBOL, FORTRAN and C are commonly known as Procedure Oriented Programming (POP) languages. POP basically consists of a list of instructions for the computer to follow and these are organised into groups, known as functions.

Tender Heart High School, Sec 33B, Chandigarh

Class IX Subject Computer Application

Topic: Introduction to Object Oriented Programming concepts

Teacher: Prabhdeep Kaur

Object oriented Programming is a type of programming which uses objects and classes in its functioning. The object oriented Programming (OOP's) is based on real world entities like inheritance, Polymorphism etc and Java is an object oriented Programming. for eg. in real life, a car is an object. The car has attributes, such as weight and colour, and methods, such as drive and brake. A class is like an object constructor, or a "blueprint" for creating objects.

OOP does not allow data to flow freely from function to function and Procedure to Procedure. In this system, the complete problem is broken into number of entities called objects. These objects are created and maintained along with a set of related data.

Features of object Oriented Programming (OOP)

* OOP restricts the free movement of data and the functions that operate on it.
* The program resulting from OOP is collection of objects
* It gives more emphasis on data rather than procedure.
* It makes the complete program simple by dividing it into a number of objects.
* The objects may communicate with each other through functions.

Basic Elements of OOP (Object Oriented Programming)

- Objects
- Classes
- Data Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Data Hiding

1. Objects

An object is an identifiable entity that has Properties for identifying its State, Methods for Behavior. Data associated at any given instance of time is the state of an object. Every Object will different from other objects either by State or behavior. The objet has the following characteristics:-

1. It has a State (instance variable)
2. It represent behaviour (instance method) such as deposit(), withdraw() etc.
3. object is an Instance of a class.
4. object has a lifetime - It is created, used and destroyed.

In OOP a complete program is split into a number of segments called objects. Each segments contains data and related methods. The data elements of one object are only accessed through methods of the same object. In this way free flowing of data is restricted throughout the program. However objects may interact with each other through the methods. An object acts as a bridge for communication

Tender Heart High School, Sec 33B, Chandigarh

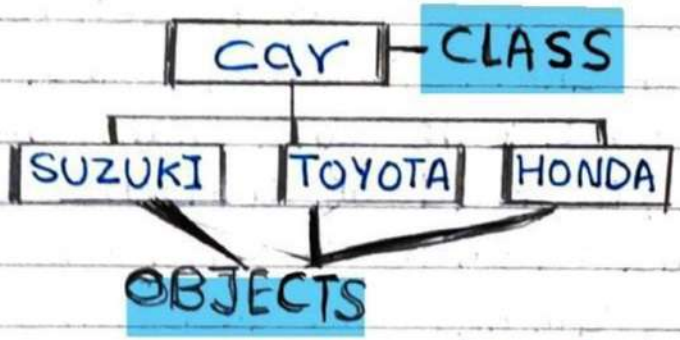
Class IX Subject Computer Application

Topic: Introduction to Object Oriented Programming concepts

Teacher: Prabhdeep Kaur

2. Class

A class is a template / blueprint for creating an object. Class is a collection of the common type of the information or we can say Class is the collection of the objects that has common properties.



With this example we got an idea of class and objects with reference to object oriented programming.

3. Data Abstraction

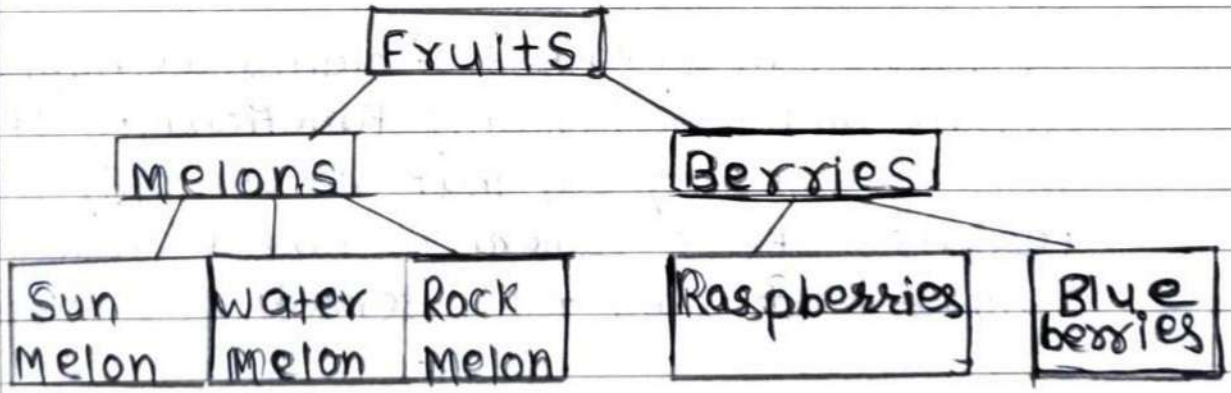
Abstraction is a process of hiding the Implementation details and showing only functionality of the user. So, we may say that we use only the essential features of the Camera to take a photograph without knowing the internal mechanism.

4 Encapsulation

Encapsulation is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. Encapsulation in OOP restrict the free flow of data. However, the data can be accessed through functions which are combined along with the Class.

5. Inheritance

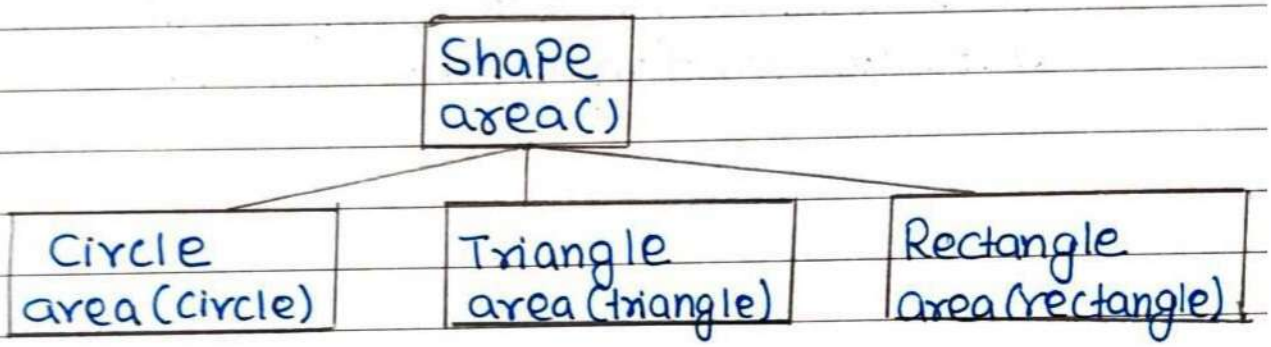
The term inheritance means to link and share some common properties of one class with other class. This can be done by extending the obects of one class into another class and using through it.
Inheritace allows us to define a class in terms of another class, which makes it easier to create and maintain. When creating a class, instead of writing completely new data members and member functions, the programmar can designate that the new class should inherit the members of an existing class. Hence, reusability is one of the important features of inheritance.



So, in this example we see that class 'fruit' which can broadly classified as Melons and Berries. ~~have~~ same properties of the class fruits will be inherited by the classes Melons and Berries

6. Polymorphism

Polymorphism means "many forms" and it occurs when we have many classes that are related to each other by inheritance. So, we can say polymorphism is the process of using a function/ ~~contact~~ method for more than one purpose.



In this example function name is areaC and we can calculate the area of different figures. Polymorphism is one of the oop principles which supports function overloading because in above example we have same function name areaC() but with different parameters.

7. Data Hiding

The process of creating logical insulation between data and outside world is known as Data Hiding.

Tender Heart High School, Sec 33B, Chandigarh

Class IX Subject Computer Application

Topic: Introduction to Object Oriented Programming concepts

Teacher: Prabhdeep Kaur

Answer the following:

Q1. What do you understand by the term data abstraction? Explain with example.

Q2 What is the difference between an object and a class?

Q3 What does reusability mean?

Q4 Why is 'data Hiding' required?

Q5 In what ways are Encapsulation and Data Abstraction inter-related?