

GIRLS' HIGH SCHOOL AND COLLEGE

2020-2021

CLASS 11 B

COMPUTER SCIENCE

WORKSHEET -01

CHAPTER- GENERAL OOP CONCEPTS

Note: Parents please ensure that your ward refers to a computer science book/internet for two days to read the topic General OOP Concepts.

Reference Book: Computer Science with Java for class 11 by Sumita Arora

Website: [geeksforgeeks.org](https://www.geeksforgeeks.org)

Introduction:

Evolution of Software

The program serves the purpose of commanding the computer. The efficiency and usefulness of a program depends not only on proper use of, commands but also on the programming language it is written in. The two major types of programming languages: Low level languages and High level languages offer different features of programming.

Low Level Language

Low level languages (i.e. machine language and Assembly language) are machine oriented and require extensive knowledge of computer circuitry. Machine language in which instructions are written in binary code (using 0 and 1) is the only language the computer can execute directly. Assembly language in which instructions are written using symbolic names for machine operations (e.g. READ, ADD, STORE etc) and operands makes program less tedious than machine language programming. However assembly language is then converted into machine language using assembler software.

High Level Language

High Level Languages (HLLs) on the other hand offer English like keywords construct for sequence selection and iteration and use of variables and constants thus it is very easy to program with such languages compared to low level languages. The programs written in HLLs are converted into machine language using compiler or interpreter as a computer can work with machine language only.

Programming Paradigms

Paradigm means organizing principle of a program. It is an approach to programming. The primary motivation in each paradigm has been the concern to handle the increasing complexity of programs that are reliable and maintainable.

Procedural Programming

A program in a procedural language is a list of instructions where each statement tells the computer to do something. The focus is on the processing, the algorithm needed to perform the desired computation.

In procedural paradigm, *the emphasis is on doing things rather on data.*

Modular Programming

In modular programming, a large program is broken down into smaller units called functions or sub-programs. The idea of breaking a program into functions can further be extended by grouping a number of functions together into a larger entity called a module.

In modular programming, since many modules access the same data, the way the data is stored becomes critical. The arrangement of the data can't be changed without modifying all the functions that access it.

A set of related procedures with the data they manipulate is called a module.

Object oriented programming

Object oriented programming is based on the principle of Data Hiding, Abstraction Encapsulation, Modularity, Inheritance and Polymorphism implements program using the

objects in an object oriented language. The object oriented approach views a problem in terms of objects involved rather than procedure for doing it.

Object represents data and its associated functions in a single unit.

A Class represents a group of similar object.

Basic Concepts of OOP

1. Data Abstraction

Abstraction refers to the act of representing essential features without including the background details or explanations. It is used for hiding the unwanted data and giving relevant data. Abstraction lets you focus on what the object does instead of how it does it.

2. Encapsulation

The wrapping up of data and functions into a single unit (called class) is known as encapsulation. It means hiding the code and data into a single unit to protect the data from outside world.

Encapsulation is a way to implement data abstraction. Encapsulation hides the details of the implementation of an object.

Abstraction and encapsulation are complementary concepts: abstraction focuses upon the observable behaviour of an object whereas encapsulation focuses upon the implementation that gives rise to this behaviour. Encapsulation is most often achieved through information hiding which is the process of hiding all the secrets of an object that do not contribute to its essential characteristics, the structure of an object is hidden, as well as the implementation of its method. Only the essential characteristics of object are visible.

3. Modularity

Modularity is the property of decomposing a system into set of cohesive and loosely coupled modules. The act of partitioning a program into individual components is called modularity.

The justification for partitioning a program is that:

- i. it reduces its complexity to some degree and

- ii. it creates a number of well defined documented boundary within the program.

4. Inheritance

Inheritance is the capability of a class of things to inherit capabilities or properties from another class. The class that inherits properties from another class is subclass or derived class and the other class is base class.

A subclass class defines only those features that are unique to it, rest it inherits from its base class.

5. Polymorphism

Polymorphism is a key to the power object oriented programming. It is the concept that supports the capability of an object of a class to behave differently in response to the message for action. *Polymorphism is the ability for a message or data to be processed in more than one form operation is performed differently depending upon the data type it is working upon.*

Answer the following questions:

1. What are the two major types of programming languages?
2. Which two programming languages are low level languages?
3. How are programs written in:
 - i. Machine language
 - ii. Assembly language
4. What do you mean by programming paradigm?
5. What are the characteristics of procedural paradigm?
6. What is a module? What is modular programming paradigm? What are its characteristics?
7. What is object oriented programming? Name the basic concepts of OOP.
8. What is an object? What is a class? How is an object different from a class?
9. What is meant by Abstraction?
10. What is Encapsulation?

11. What is a base class? What is a subclass? What is a relationship between a base class and a subclass?
12. What is modularity? What benefits does it offer?
13. What is Polymorphism? Give an example to illustrate polymorphism?

-----END-----