info@marwadiuniversity.ac.in

Marwadi University

Index

Name.	Shashank Bagda	Year.	2021/22
Subject.	OOP	Class.	2TK1
Semester	2	Roll No.	92100133020

No.	Date	Experiment Description / Subject	Page	Marks	Faculty's Signature
1	3/3/22	Worksheet - 1			[illegible]

Subject Code: 01CT0105

Subject Name: Object Oriented Programming

B. Tech. Year – I (Semester II)

UNIT – 1

Worksheet – 1

Enrollment No: 92100133020

Name: Shashank Bagda

Subject Faculty:

Prof. Kapil Shukla

Information and Communication Technology Department,

Marwadi University

Q-1: Explain following terms with example.

Answer:

a) Object Oriented:

The organising software as a collection of discrete object that incorporate both data structure or attribute and behaviour. This contrasts with previous programming approaches in which data structure & behaviour are only loosely connected.

b) Identity:

A property of data that is created in context of an object data model, where an object is assigned a unique internal object identifier or place where its value is stored in memory.

c) Object:

Identity means that data is quantized into discrete, distinguishable entities called object. Objects can be concrete, such as a file in a file system, or conceptual, such as a scheduling policy in a multiprocessing operating system. Each object has its own inherent identity.

d) Classification:

Classification means that objects with the same data structure and behaviour are grouped into a class. Paragraph, Monitor & Chess Piece are example of classes. A class is an abstraction that describes properties important to an application and ignores the rest.

e) Attribute:

In object oriented programming, class & object have attributes. Attributes are data stored inside a class or instance and represent the state or quality of the class or instance. In short attributes store information about the instance.

f) Operation:

An operation is a service that can be requested from any object of the class to affect behaviour. An operation can either be a command or a question. A question should never change the state of the object only a command can.

g) Instance:

Each class describes a possibly infinite set of individual objects. Each object is said to be an instance of its class. An object has its own value for each attribute but shares the attribute names and operations with other instances of the class.

h) Inheritance:

Inheritance is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents with inheritance we can reuse the fields and methods of the existing class.

i) Superclass and Subclass:

A superclass has general information that subclass refine and elaborate. Each subclass incorporates or inherits all the features of the superclass and adds its own unique features. Subclasses need not repeat the feature of the superclass & its own unique feature.

j) Polymorphism:

Polymorophism referos to the ability of a variable, function oro object to take on multiple foroms. In proogroamming language exibiting polymorophism, class objects belonging to the same hienachical troee may have function with the same name but different behaviouro.

k) Operation:

An operoation is a procedure oro troansforomation that an object peroforoms oro is subject to right justify, displays and move are example of operations.

l) Method:

A method is a collection of statements that peroforom some specific task and return results to the callero. A method can peroforom some specific task without returoning anything. methods are time savers and help us to reuse the code without retyphy the code.

Q-2: Explain OO Methodology stages.

Answer: The methodology have following stages :

1) System Conception :

→ Software development begins with business analysts oro useros conceiving an application and foromulating tentative requirements.

2) Analysis :

→ The analyst restats the requirements from system conception by constructing models. The analyst must work with the requestoro to underostand the problem, because problem statements are rarely complete oro coroorect. The analysis model is a concise abstraction of what the desired system must do, now how it will be done. The analysis model should not contain impletation decisions. For example, a window class in a workstation windowing system would be described in terms of its visible attributes & operations.

The analysis model has two parts; the domain model, a description of the real world object reflected within the system · and the application model, a description of the parts of the application system itself that are visible to the user. For example, domain objects for a stockbroker application might include stock, bond, trade and commission. Application objects might control the execution of trades and present the results. Application experts who are not programmers can understand.

Q-3: What is the difference between Domain Model and Application Model.

Answer:

Domain Model	Application Model
• A description of the real world objects reflected within the system.	• A description of the parts of the application system itself that are visible to users.
• When we know every thing about the main thing.	• If we know all things, We can implement what we need.
Eg: If we read whole chapter of a subject.	Eg: If we write only what is asked in exam.

Q-4: Explain three models.

Answer: We use three kinds of models to describe a system from different viewpoints: the class models for the objects in the system and their relationship the state model for the life history of objects and the interaction model for the interaction among objects; Each model applies during all stages of development and acquires detail as development progresses. A complete description of a system requires model from all three viewpoints.

The class model describes the static structure of the objects in a system and their relationship. The class model defines the context for software development. the universe of discourse. The class model contains class diagram.

A class diagram is a graph whose nodes are classes and whose arcs are relationships among classes. The state model describes the aspects of an object that change over time. The state model specifies and implement control with the state diagrams is a graph whose nodes are states and whose arcs are transitions between state caused by events. ~~The~~ interaction model describes how the objects in a system cooperate to achieve broader results.

The interaction model starts with use cases that are then elaborated with sequence and activity diagrams. A use case focuses on the functionality of a system, that is what a system does for users. A sequence diagram shows the objects that interact and the time sequence of their interactions. An activity diagram elaborates important processing steps. The three models are separate parts of the ~~diagra.~~ description of a complete system but ~~or~~ are cross-linked. The class model is ~~not~~ most fundamental, because it is necessary to describe what is changing or transforming before describing when or how it changes.

Q-5: Explain OO Themes.

Answer: ① Abstroaction

Abstroaction will let you focus on essential aspects of an application while ignoroing detapls. This means focusing on what an object is and does before deciding how to implement it, use of abstroaction proeserves the freedom to make decision as long as possible by avoiding premature commitments to details.

② Encapsulation

Encapsulation separoats the exteronal aspects of an object, that are accessible to othero objects, from the interonal implementation detail, that are hidden from others objects. It prevents parotions of a proogroam from becoming so Interodependent that a small change has massive ripple effects.

③ Combining data & Behaviouro

The callero of an operation need not consider how many implementations exist. Operator polymorophism shifts the buroden of deciding what implementation to use from the calling code to the class hierarchy.

④ Sharing

Inheritance of both data structure and behaviour lets subclasses share common code. This sharing via inheritance is one of the main advantages of OO language.
More important than the savings In code is the conceptual clarity from recognizing that different operations are all really the same thing.

(5) Emphasis on the Essence of an Object.

The uses of the object depend on the details of the application and often change during development, As requirements evolve, the features supplied by an object are much more stable than the ways its used, hence software systems built on object structure are more stable in the long run.

(6) Synergy !

Identity, classification, polymorphism and inheritance characterize OO languages. Each of these concepts can be used in isolation, but together they complement each other synergistically. The benefits of an OO approach are greater than they might seem at first. The emphasis on the essential properties of an object forces the developer to think more carefully and deeply about What an object is and does. The resulting system tends to be cleaner, more general, and more robust than it would be if the emphasis were only on the use of data and operation.

[illegible]

References

Q No	Book Name	Page No
1	Study Materoial	1, 2
2	Study Materoial & OO Modeling & Design	4
3	Study Materoial & OO Modelling & Design	5
4	Study Materoial & OO Modeling & Design	6
5	Study Materoial	6, 7, 8