

K2 Level -Unit 3 Object Oriented Analysis, Structured analysis and structured design (SA/SD)

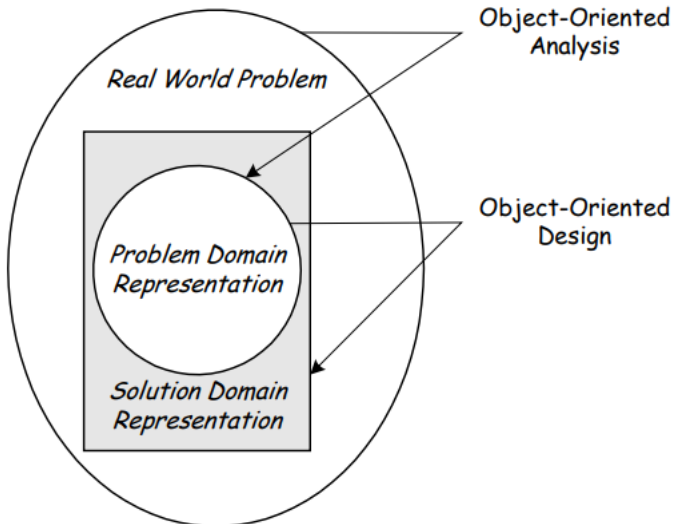
Subjective Question -Unit 3 Object Oriented Analysis, Structured analysis and structured design (SA/SD)	
Question:1	Describe object oriented analysis concepts.
Solution: 1 (With step wise marking)	<p>Object-Oriented Analysis (OOA) is the first technical activity performed as part of object-oriented software engineering. OOA introduces new concepts to investigate a problem. It is based on a set of basic principles, which are as follows-</p> <ol style="list-style-type: none"> 1. The information domain is modeled. 2. Behavior is represented. 3. The function is described. 4. Data, functional, and behavioral models are divided to uncover greater detail.

Question:2	Differentiate between structured analysis and object oriented analysis.
Solution: 2 (With step wise marking)	<ol style="list-style-type: none"> 1. <u>Structured Analysis :</u> Structured analysis is a method of development that allows and gives permission to the analyst to understand and know about the system and all of its activities in a logical way. It is simply a graphic that is used to specify the presentation of the application. 2. <u>Object-Oriented Analysis :</u> Object-Oriented Analysis (OOA) is a technical approach generally used for analyzing and application designing, system designing, or even business designing just by applying object-oriented programming even with the use of visual modeling throughout the process of development to just simply guide the stakeholder communication and quality of the product. it is actually a process of discovery where a team of developers understands and models all the requirements of the system.

Question:3	Describe a class for a system using a suitable diagram.
-------------------	---

<p>Solution: 3 (With step wise marking)</p>	<p>A class is a blueprint or prototype that defines the variables and methods common to all objects of a certain kind.</p> <ul style="list-style-type: none"> • In the real world, we often have many objects of the same kind. For example, our bicycle is just one of many bicycles in the world. Using object-oriented terminology, we say that our bicycle object is an instance of the class of objects known as bicycles. Bicycles have some attribute (current gear, current cadence, two wheels) and behavior (change gears, brake) in common. However, each bicycle's attribute is independent of and can be different from other bicycles. • When building bicycles, manufacturers take advantage of the fact that bicycles share characteristics by building many bicycles from the same blueprint. It would be very inefficient to produce a new blueprint for every individual bicycle they manufactured.
--	--

Table 1

<p>Question:4</p>	<p>Discuss the relationship between object oriented analysis and object-oriented design.</p>
<p>Solution: 4 (With step wise marking)</p>	<p><u>Relationship between OOA and OOD</u></p> <ul style="list-style-type: none"> • OOA involves problem definition and design focuses on solution specification. • OOD transforms the problem representation into a solution representation. • The problem and solution domain representations are different and smaller than real-world problem. • And the solution domain includes everything in the problem domain, plus any additional constructs required by the solution. • However, it is difficult to determine where OOA ends and OOD begins, because of the blurred distinction between analysis and design in the object paradigm. 

	<p>OOA is concerned with devising a precise, concise, understandable, and correct model of real-world. Before building anything complex, such as a house, a computer program, or hardware-software system, the builder must understand the requirements and the real-world environment in which will exist.</p> <p><u>Purpose of OOA</u> is to model the real-world system so that it can be understood. To do this, we must examine requirements, analyze their implications, and restate them rigorously. We must abstract important real-world features first and defer small details until later. The successful analysis model states what must be done, without restricting how it is done, and avoids implementation decisions.</p> <p><u>Purpose of OOD</u> The product of OOA is used to make design further in OOD.</p>
--	---

Question:5	Discuss the various steps in Object-oriented analysis and design process using a suitable diagram.
Solution: 5 (With step wise marking)	<p><u>Object oriented and analysis process</u></p> <ul style="list-style-type: none"> • OOAD begins with a problem statement (requirement) generated by users and possibly customer. • The requirement may be incomplete or informal, and identification processes make it more precise and exposes ambiguities and inconsistencies. • The requirement should not be taken as immutable but should serve as a basis for refining the real requirements. Next, the real-world system described by the requirement must be understood and identified, and its essential features abstracted into a model. • Identifying objects, attributes, associations and behaviours of the object is the important step in constructing an object model. • And the next step is to organize classes by using inheritance to share common structure. • Inheritance can be added in two directions: <ol style="list-style-type: none"> 1. by generalizing common aspects of existing classes into a superclass (bottom up or generalization) 2. refining existing classes into specialized subclasses (top down or specialization). • The object identification and refinement process are called object model creation process. • The last OOAD step is to implement class model using a programming language.

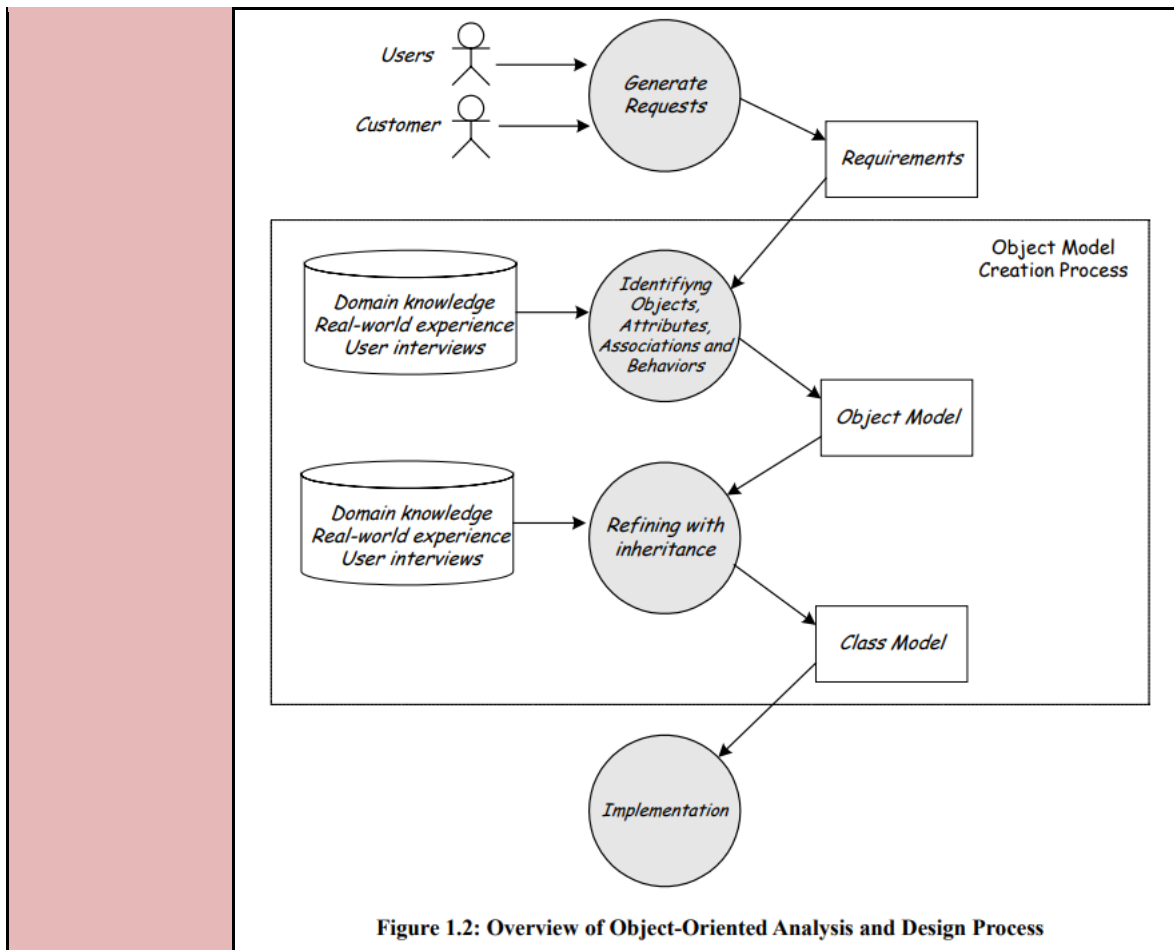
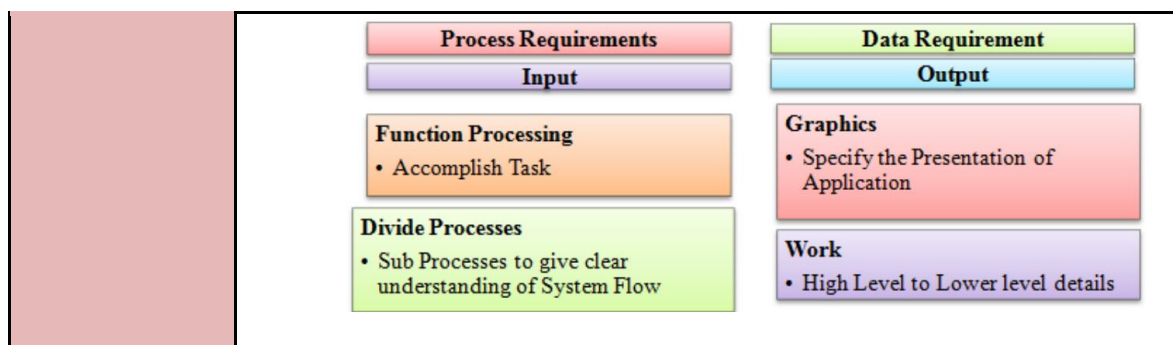


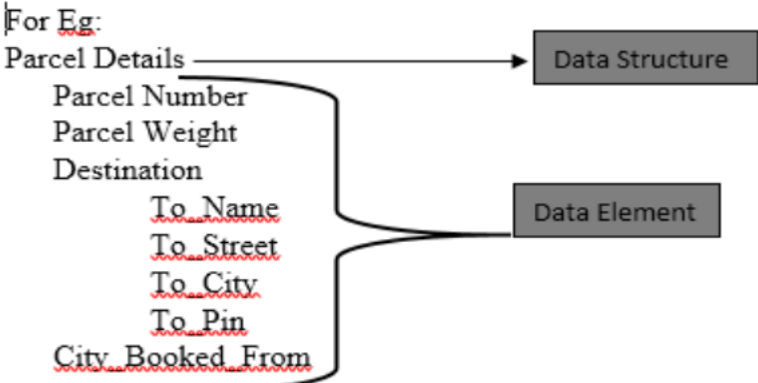
Table 2

Question:6	Describe various steps in structured analysis.
Solution: 6 (With step wise marking)	<p>Structured Analysis is a logical tool to understand and describe the information system in a logical way which uses graphical tools in an organized manner that makes use of graphical diagrams to develop and present system specifications to users in a way that makes them clear and easy to understand.</p> <p>Some of the important steps involved in structured analysis are as below:</p> <ul style="list-style-type: none"> • Studying the current system and evaluating all of its issues • Modelling this system • Modelling the new system around these issues, in order to fix them • Modelling the new physical environment • Evaluating any alternatives • Choosing the best system approach • Creating the graphical specifications



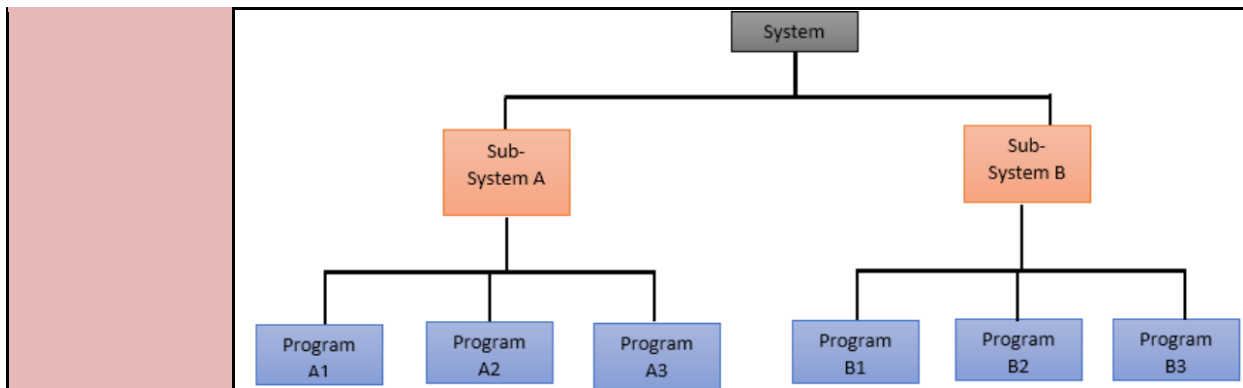
Question:7	Discuss various tools of structured analysis.
Solution: 7 (With step wise marking)	<p>Tools of Structured Analysis:</p> <p>Following are the various tools and techniques which are used for system development. They are –</p> <ul style="list-style-type: none"> • ER Model • Flowchart • Data Flow Diagrams • Decision Trees • Decision Tables • Structured English • Data Dictionary

Question:8	Describe data dictionary as tools for structured analysis. Give a suitable example to illustrate data dictionary.
Solution: 8 (With step wise marking)	<p>Data Dictionary:</p> <p>Defines each term (element) encountered during the analysis and design of a new system. It is a special kind of dictionary which contains all the information about the data of a system. It forms an integral part of structured specification. It is only a documentation of data.</p> <p>The important components of data dictionary are:</p> <ol style="list-style-type: none"> 1. Data Element: Smallest unit of data that is meaningful or pieces of data which can't be meaningfully decomposed are known as data elements. For each data element the data dictionary should hold the following description: <ul style="list-style-type: none"> • Name • Description • Name of the relative data element • Length and type of data element • Codification Structure • Range of value and their meaning 2. Data Structure: They are made up data element or a combination of both data element and data structure. They should contain the following information: <ul style="list-style-type: none"> • Name • Description

	<ul style="list-style-type: none"> • Included data elements and data structure with brief description <p>3. Data Flow: Contains the volume of flow (Start, End) point with the description of data.</p> <p>4. Data Store: Additional information on the volume and data flows to and from it is also usually recorded.</p> <p>5. Process: Documenting the processes include documenting name and brief description of the process followed by a process description. The entry for each and every process forms a part of data dictionary.</p> <p>For Eg:</p>  <p>Fig 1.10 Example of Data Dictionary</p>
--	--

Question:9	Discuss the importance of structured chart in structured analysis.
Solution: 9 (With step wise marking)	<p>Structured Chart:</p> <p>A top-down modular design tool which shows the breakdown of a system to its lowest manageable levels with the help of graphical representation of:</p> <ul style="list-style-type: none"> • Squares represent different modules in the system • lines that connect modules which shows the relationship between modules. <p>A structure chart is a tree of sub-routines in a program which indicates the interconnections among the sub-routines. The sub-routines should be labeled with the same name used in the pseudo code.</p>

Question:10	Explain structured chart using a diagram.
Solution: 10 (With step wise marking)	<p>A structure chart depicts:</p> <ul style="list-style-type: none"> • the size and complexity of the system, and • number of readily identifiable functions and modules within each function and • Whether each identifiable function is a manageable entity or should be broken down into smaller components.



Question:11	Explain Jackson system development approach for developing a system.
Solution: 11 (With step wise marking)	<p><u>Jackson System Development (JSD)</u></p> <p>It is a method of system development that covers the software life cycle either directly or by providing a framework into which more specialized techniques can fit. JSD can start from the stage in a project when there is only a general statement of requirements. However many projects that have used JSD actually started slightly later in the life cycle, doing the first steps largely from existing documents rather than directly with the users.</p> <p><u>Phases of JSD:</u></p> <p>JSD has 3 phases:</p> <ol style="list-style-type: none"> 1. Modelling Phase: In the modelling phase of JSD, the designer creates a collection of entity structure diagrams and identifies the entities in the system, the actions they perform, the attributes of the actions and the time order of the actions in the life of the entities. 2. Specification Phase: This phase focuses on actually what is to be done? Previous phase provides the basic for this phase. An sufficient model of a time-ordered world must itself be time-ordered. Major goal is to map progress in the real world on progress in the system that models it. 3. Implementation Phase: In the implementation phase JSD determines how to obtain the required functionality. Implementation way of the system is based on the transformation of the specification into an efficient set of processes. The processes involved in it should be designed in such a manner that it would be possible to run them on available software and hardware.

Question:12	Discuss merits of Jackson system development.
--------------------	---

Solution: 12 (With step wise marking)	Merits of JSD: <ul style="list-style-type: none"> • It is designed to solve the real-time problems. • JSD modelling focuses on time. • It considers simultaneous processing and timing. • Provides functionality in the real world. • It is a better approach for microcode applications.
--	---

Question:13	Discuss demerits of Jackson system development.
Solution: 13 (With step wise marking)	Demerits of JSD: <ul style="list-style-type: none"> • It is a poor methodology for high level analysis and database design. • JSD is a complex methodology due to pseudo code representation. • It is less graphically oriented as compared to SA/SD or OMT. • It is a bit complex and difficult to understand.

Question:14	Describe concept of object in OOPs?
Solution: 14 (With step wise marking)	<p>A class instance is referred to as an object. A real-world object is something like a pen, a laptop, a phone, a bed, a keyboard, a mouse, or a chair. A physical entity is referred to as an object. In Java, there are many methods for creating objects, including the new keyword, the newInstance() method, the clone() method, the factory method, and deserialization.</p> <p>Here is the real-life example of an object in OOPS is: Class: Human, Object: Man, Woman Class: Fruit, Object: Apple, Banana, Mango, Guava, etc.</p>

Question:15	Which are the best features of OOPs, and why explain?
Solution: 15 (With step wise marking)	<p>Encapsulation, inheritance, and polymorphism are three fundamental elements of object-oriented programming that distinguish it from non-OOP languages.</p> <p>Encapsulation is the process of creating self-contained modules that connect processing processes to data. Classes are organized into hierarchies, and inheritance enables one class's structure and functions to be transferred down the hierarchy. Finally, object-oriented programming enables the creation of procedures regarding objects whose precise type is unknown until runtime is known as polymorphism.</p>

Table 3

Question:16	Differentiate procedural and object-oriented programming language.	
Solution: 16 (With step wise marking)	<u>Procedural vs OOP</u>	
	procedural Oriented Programming	Object-Oriented Programming
	In procedural programming, the program is divided into small parts called functions .	In object-oriented programming, the program is divided into small parts called objects .
	Procedural programming follows a top-down approach .	Object-oriented programming follows a bottom-up approach .
	There is no access specifier in procedural programming.	Object-oriented programming has access specifiers like private, public, protected, etc.
	Adding new data and functions is not easy.	Adding new data and function is easy.
	Procedural programming does not have any proper way of hiding data so it is less secure .	Object-oriented programming provides data hiding so it is more secure .
	In procedural programming, overloading is not possible.	Overloading is possible in object-oriented programming.
	In procedural programming, there is no concept of data hiding and inheritance.	In object-oriented programming, the concept of data hiding and inheritance is used.
	In procedural programming, the function is more important than the data.	In object-oriented programming, data is more important than function.
	Procedural programming is based on the unreal world .	Object-oriented programming is based on the real world .

	Procedural programming is used for designing medium-sized programs.	Object-oriented programming is used for designing large and complex programs.
	Procedural programming uses the concept of procedure abstraction.	Object-oriented programming uses the concept of data abstraction.
	Code reusability absent in procedural programming,	Code reusability present in object-oriented programming.
	Examples: C, FORTRAN, Pascal, Basic, etc.	Examples: C++, Java, Python, C#, etc.

Question:17	Explain the mapping steps of object-oriented concepts using a non-object oriented language. Give a suitable example.
Solution: 17 (With step wise marking)	<p>Implementing an object-oriented concept in a non-object oriented language requires the following steps :</p> <p>1. Translate classes into data structures :</p> <p>i. Each class is implemented as a single contiguous block of attributes. Each attribute contains a variable. Now an object has state and identity and is subject to side effects.</p> <p>ii. A variable that identifies an object must therefore be implemented as a sharable reference.</p> <p>2. Pass arguments to methods :</p> <p>i. Every method has at least one argument. In a non-object-oriented language, the argument must be made explicit.</p> <p>ii. Methods can contain additional objects as arguments. In passing an object as an argument to a method, a reference to the object must be passed if the value of the object can be updated within the method.</p> <p>3. Allocate storage for objects :</p> <p>i. Objects can be allocated statically, dynamically or on a stack.</p> <p>ii. Most temporary and intermediate objects are implemented as stack-based variables.</p> <p>iii. Dynamically allocated objects are used when there number is not known at compile time.</p> <p>iv. A general object can be implemented as a data structure allocated on request at run time from a heap.</p> <p>4 Implement inheritance in data structures :Following ways are use to implement data structure for inheritance in non object oriented programming language</p> <ul style="list-style-type: none"> • Avoid it. • Flatten the class hierarchy.

	<ul style="list-style-type: none"> • Break out separate objects. <p>5 Implement method resolution : Method resolution is one. main features of an object-oriented language that is lacking in a non object-oriented language. Method resolution can be implemented in a following ways</p> <ul style="list-style-type: none"> • Avoid it. • Resolve methods at compile time. • Resolve methods at run time. <p>6. Implement associations : Implementing associations in a non oriented language can be done by :</p> <ul style="list-style-type: none"> • Mapping them into pointers. • Implementing them directly as association container objects. <p>7. Deal with concurrency :</p> <ul style="list-style-type: none"> • Most languages do not explicitly support concurrency. • Concurrency is usually needed only when more than one external event occurs, and the behaviour of the program depends on their timing. <p>8. Encapsulate internal details of classes :</p> <ul style="list-style-type: none"> • Object-oriented languages provide constructs to encapsulate implementation. • Some of this encapsulation is lost when object-oriented concepts translated into a non-object-oriented language, but we can still take advantage of the encapsulation facilities provided by language
--	---

Question:18	Explain inline function working .
Solution: 18 (With step wise marking)	<p>In inline functions, there is no function calls, the compiler replace every instance of their calls in the code with the body of the function saving the overhead associated with the function call mechanism.</p> <p>- "inline" keyword can be used to specify a function as an inline.</p>

Question:19	Where is the memory allocated for the objects?
Solution: 19 (With step wise marking)	<ul style="list-style-type: none"> • Many times, you are not aware in advance how much memory you will need to store particular information in a defined variable and the size of required memory can be determined at run time. • You can allocate memory at run time within the heap for the variable of a given type using a special operator in C++ which returns the address of the space allocated. This operator is called new operator.

	<ul style="list-style-type: none"> If you are not in need of dynamically allocated memory anymore, you can use delete operator, which de-allocates memory previously allocated by new operator.
--	---

Question:20	Differentiate between SA/SD and OMT approach.
Solution: 20 (With step wise marking)	<ul style="list-style-type: none"> In the SA/SD approach, the functional model dominates, the dynamic model is next most important, the object model least important. In contrary to this, OMT modelling regards the object model as most important, then the dynamic model, finally the functional model. The Structured Analysis/Structured Design approach organizes a system around procedures. On the contrary, an object oriented design technique organizes a system around real world objects, or conceptual objects that exist in the user's view of the world. Most changes in requirements are changes in function rather than in objects, so change can be disastrous to procedure based design. By contrast, changes in function are readily accommodated in an object-oriented design by adding or changing operations, leaving the basic object structure unchanged. An SA/SD design has a clearly defined system boundary, across which the software procedures must communicate with the real world. The structure of a SA/SD design is derived in part from the system boundary, so it can be difficult to extend a SA/SD design to a new boundary. To the contrary, it is much easier to extend an object-oriented design. <p>In SA/SD the decomposition of a process into sub-processes is somewhat arbitrary. Different people will produce different decompositions. In the object-oriented design the decomposition is based on objects in the problem domain, so developers of different programs in the same domain tend to discover similar objects. This increases reusability of components from one project to the next.</p> <p>The object-oriented approach better integrates databases with the programming code. One common uniform paradigm, the object, can model both database and programming structure. In contrast, a procedural design approach is inherently awkward at dealing with databases.</p>

Question:21	Discuss the concept of concurrency in OO analysis.
Solution: 21 (With step wise marking)	For certain kinds of problems, an automated system may have to handle many different events simultaneously. Other problems may involve so much computation that they exceed the capacity of any single processor. In each of these cases, it is natural to consider using a distributed set of computers for

	the target implementation or to use processors capable of multitasking. A single process - also known as a thread of control is the root from which independent dynamic action occurs within a system. Every program has at least one thread of control, but a system involving concurrency may have many such threads: some that are transitory, and others that last the entire lifetime of the system's execution. Systems executing across multiple CPUs allow for truly concurrent threads of control, whereas systems running on a single CPU can only achieve the illusion of concurrent threads of control, usually by means of some time-slicing algorithm.
--	--

Question:22	Why Software Is Inherently Complex?
Solution: 22 (With step wise marking)	This external complexity usually springs from the "impedance mismatch" that exists between the users of a system and its developers: users generally find it very hard to give precise expression to their needs in a form that developers can understand. In extreme cases, users may have only vague ideas of what they want in a software system. This is not so much the fault of either the users or the developers of a system; rather, it occurs because each group generally lacks expertise in the domain of the other. Users and developers have different perspectives on the nature of the problem and make different assumptions regarding the nature of the solution.

Table 4

Question:23	Explain the concept of persistence in object oriented system.
Solution: 24 (With step wise marking)	<p>An object in software takes up some amount of space and exists for a particular amount of time. Atkinson et al. suggest that there is a continuum of object existence, ranging from transitory objects that arise within the evaluation of an expression, to objects in a database that outlive the execution of a single program.</p> <p>This spectrum of object persistence encompasses the following:</p> <ul style="list-style-type: none"> ○ “Transient results in expression evaluation ○ Local variables in procedure activations ○ Own variables global variables, and heap items whose extent is different from their scope ○ Data that exists between executions of a program ○ Data that exists between various versions of a program ○ Data that outlives the program"

Question:25	How invoking method is different in traditional and object oriented programming?
Solution: 25 (With step wise marking)	In traditional programming languages, invoking a subprogram is a completely static activity. In Pascal for example, for a statement that calls the subprogram P, a compiler will typically generate code that creates a new stack frame, places the proper arguments on the stack, and then changes the flow of control to begin executing the code associated with P. However, in

	languages that support some form of polymorphism, such as Smalltalk, and C++, invoking an operation may require a dynamic activity, because the class of the object being operated upon may not be known until runtime. Matters are even more interesting when we add inheritance to the situation. The semantics of invoking an operation in the presence of inheritance without polymorphism is largely the same as for a simple static subprogram call, but in the presence of polymorphism, we must use a much more sophisticated technique.
--	--

Table 5

Question:26	Explain the role of classes and objects in analysis and design.
Solution: 26 (With step wise marking)	classes and objects are the key abstractions of the problem, and these are cooperative structures the mechanisms of the implementation. During these phases of development, the focus of the developer must be upon the outside view of these key abstractions and mechanisms. This view represents the logical framework of the system, and therefore encompasses the class structure and object structure of the system. In the later stages of design and then moving into implementation, the task of the developer changes: the focus is on the inside view of these key abstractions and mechanisms, involving their physical representation.

Question:27	Describe data abstractions in brief using example.
Solution: 27 (With step wise marking)	Data abstraction- Creating a structure that defines a complex data type and legal operations on it. The design makes the interface of the data and operations accessible but hides the actual implementation of them. Example: A List class can choose to implement the container of the data elements as an array or as a linked list. This container is kept hidden from the user interface but public interface is provided to element read/write operations.