Chapter 6 - Object Oriented Concepts

OBJECT ORIENTED CONCEPTS

Introduction

- The programming languages deal with two important aspects – data and algorithm.
- Data is the main substance of the program and the algorithms are the methods that use these programs.
- OOP is the principle of design and development of programs using the modular approach.

Procedural programming:

- The procedural programming focuses on processing of instructions in order to perform a desired computation. Therefore it emphasizes more on doing things like algorithms.
- This programming is lengthy, increases the complexity of program, difficult to understand and modify the program.
- This technique is used in conventional programming languages such as C and Pascal.
- For example, procedural programs make use of branching statements, where the flow of control changes its path according to the result of the test expression.

Structured programming:

- Structured programming approach was a disciplined approach which limits the branching to a small set of well-behaved construction (ex: if, if-else, while etc).
- The idea of Top-down design is to break a large program into smaller programs.
- The major drawback is that it is very difficult to model the real world scenario using this model.

Object oriented programming:

- Object oriented programming (OOP) is a concept that combines both the data and the functions that operate on that data into a single unit called the object.
- Object is an identifiable entity with some characteristics and behavior.
- Top-down design is an approach of dividing a problem into sub problems and then dividing the sub problems further into still smaller sub problems until it can be implemented for a computer solution.
- Bottom up design is vice versa where solutions to smaller modules are integrated to find the
solution of overall problem.

- OOP follows bottom-up design technique.
- Class is the major concept that plays important role in this approach. Class is a template that represents a group of objects which share common properties and relationships.

➢ OOPs characteristics:

The general concepts of OOPs includes

1. Modularity:
   - Module is a logically self contained unit that can be tested and executed independently.
   - Modularity is a technique adopted to divide a complex problem into a number of self contained independent programming fragments or modules.

2. Abstraction:
   - Abstraction is an act which represents the essential features of an entity without including explanations or any background details about it.

3. Data Encapsulation:
   - Wrapping of data and functions into a single unit is called data encapsulation.
   - The concept of insulating the data from direct access by the program is called data hiding.

4. Inheritance:
   - Inheritance is the process by which objects of one class acquires the properties of the objects of another class. (Or)
   - The process of deriving a new class from the existing class is called inheritance.

5. Polymorphism:
   - Polymorphism is the ability for a message to be processed in more than one form.
   - The process of making an operator to exhibit different behaviors in different instances is known as operator overloading.
   - Using a single function name to perform different types of tasks is known as function
overloading.

6. Dynamic Binding:
   - Binding means linking of a procedure call to the code to be executed when it is called.
   - *Dynamic binding means binding the procedure call during program runtime.*

7. Message Passing:
   - *Passing message objects and invoking the function by the object by sending a message is known as message passing.*

➢ OOPs Benefits:
1. OOPs model the real world entity very well.
2. Inheritance eliminates the redundancy (repetition) of code and hence supports code reusability.
3. Data hiding helps to build secured programs.
4. Multiple instances (objects) can be created.
5. Work can be divided easily.
6. OOPs can be easily upgraded from small to large systems.
7. Complexity can be easily managed.
8. Message passing concept helps the objects to communicate and share data.

➢ OOPs disadvantages:
1. OOPs use tricky method to do the programming.
2. Special skills such as thinking in terms of design skills, programming skills and object is required for a programmer.
3. Proper planning and design is required before programming using OOPs technique.

➢ OOPs Applications:
1. Object oriented databases.
2. Hypermedia, expert text and hypertext.
3. Artificial intelligence and expert systems.
4. Decision support systems and office automation systems.
5. Parallel programming and neural networks.
6. CAD, CAM, CIM systems.
7. Simulation and modeling.

<table>
<thead>
<tr>
<th>CHAPTER 6 – OBJECT ORIENTED CONCEPTS BLUE PRINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSA (1 marks)</td>
</tr>
<tr>
<td>-</td>
</tr>
</tbody>
</table>

***************