

BT-205 Basic Computer Engineering Unit-III Notes

UNIT-III TOPICS

- Objects and Classes
- Scope Resolution Operator
- Constructors and Destructors
- Friend Functions
- Inheritance
- Polymorphism
- Function and Operator Overloading
- Types of Inheritance
- Virtual Functions
- Introduction to Data Structures

1. OBJECT ORIENTED PROGRAMMING (OOP)

OOP is programming approach based on objects and classes.

Main Features:

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

Advantages:

- Code reusability
- Better security
- Easy maintenance

2. CLASS

Class is user-defined data type that contains data members and member functions.

Syntax:

```
class Student
{
    int roll;
    public:
    void display();
};
```

3. OBJECT

Object is instance of class.

Example:

```
Student s1;
```

Advantages of Object:

- Represents real-world entity
- Access class members

4. SCOPE RESOLUTION OPERATOR (::)

Scope resolution operator is used to define member function outside class.

Example:

```
void Student::display()
{
```

```
cout << "Hello";  
}
```

Uses:

- Access global variable
- Define class functions outside class

5. CONSTRUCTORS

Constructor is special member function automatically called when object is created.

Characteristics:

- Same name as class
- No return type
- Automatically invoked

Types of Constructors:

- Default constructor
- Parameterized constructor
- Copy constructor

Example:

```
class A  
{  
public:  
A()  
{  
cout << "Constructor";  
}  
};
```

6. DESTRUCTORS

Destructor is special member function used to destroy object.

Characteristics:

- Same name as class with ~ symbol
- No arguments
- No return type

Example:

```
~A()  
{  
cout << "Destructor";  
}
```

7. FRIEND FUNCTION

Friend function can access private members of class.

Syntax:

```
friend void show();
```

Advantages:

- Access private data
- Increase flexibility

8. INHERITANCE

Inheritance is process of deriving new class from existing class.

Advantages:

- Code reuse
- Hierarchical classification

Syntax:

```
class B : public A
{
};
```

9. TYPES OF INHERITANCE

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

10. POLYMORPHISM

Polymorphism means one function behaving differently in different situations.

Types:

- Compile time polymorphism
- Run time polymorphism

11. FUNCTION OVERLOADING

Same function name with different arguments.

Example:

```
int add(int a,int b);
float add(float a,float b);
```

Advantages:

- Improves readability
- Code reuse

12. OPERATOR OVERLOADING

Operator overloading gives special meaning to operators.

Example:

Operator + can be overloaded.

Advantages:

- Easy operations on objects
- Better readability

13. VIRTUAL FUNCTION

Virtual function is member function redefined in derived class.

Syntax:

```
virtual void display();
```

Advantages:

- Supports runtime polymorphism
- Dynamic binding

14. DATA STRUCTURE

Data structure is method of organizing and storing data efficiently.

Types:

- Linear Data Structure
- Non-linear Data Structure

Examples:

- Array
- Stack
- Queue
- Linked List
- Tree
- Graph

15. ARRAY

Array stores collection of similar elements.

Advantages:

- Fast access
- Easy implementation

16. STACK

Stack follows LIFO principle.

Operations:

- Push
- Pop

Applications:

- Function calls
- Expression evaluation

17. QUEUE

Queue follows FIFO principle.

Operations:

- Insert
- Delete

Applications:

- Scheduling
- Printer queue

MOST IMPORTANT 14 MARK QUESTIONS

1. Explain objects and classes with examples.
2. Explain constructors and destructors in detail.
3. Explain inheritance and its types with diagrams.
4. Explain polymorphism and virtual functions.
5. Explain function overloading and operator overloading.
6. Explain friend function with suitable example.
7. Explain scope resolution operator with example.
8. Explain introduction to data structures and types.

9. Explain stack and queue with applications.
10. Explain runtime and compile time polymorphism.

IMPORTANT 7 MARK QUESTIONS

1. Define class and object.
2. Explain default constructor.
3. Explain copy constructor.
4. Explain destructor.
5. Explain friend function.
6. Explain single inheritance.
7. Explain multilevel inheritance.
8. Explain virtual function.
9. Explain stack and queue.
10. Explain operator overloading.

EXAM TIPS

- Learn syntax of constructors carefully.
- Practice inheritance diagrams regularly.
- Revise OOP concepts properly.
- Focus on polymorphism and overloading.
- Practice basic programs on classes and objects.