# ASSIGNMENT -4

**Assignment objectives: -**

1. Classes
2. Objects
3. Inheritance
4. Polymorphism
5. Interfaces

1. Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows:

Name Year of joining Address

Robert 1994 64C- WallsStreat

Sam 2000 68D- WallsStreat

John 1999 26B- WallsStreat

2. Create a class by name called personcls. Populate the person class with the following properties to store the information

- Firstname
- Lastname
- DOB

Create a constructor and initialize properties dynamically

Create Method by name PrintDetails which person details

1. Whether or not today is the person’s birthday

2. Check whether or not person is over 18

3. Print appropriate message

3. Create class BaseCls with a constructor having int parameter

and prints the

same using console.writeline

Create class DerivedCls with constructor having int parameter

and prints the number using console.writeline

From main create instance of only derived class and print the

output of

both constructors (hint use inheritance and base keyword)

**4. Method Overriding with Inheritance**

- Implement a base class Animal with a method MakeSound().
- Derive classes Dog and Cat, and override MakeSound() to provide different outputs.
- Use the overridden methods in the derived classes.

**5. Multilevel Inheritance**

- Define a Person class with properties Name and Age.
- Create a derived class Employee with a Salary property.
- Further extend Manager from Employee and add a Bonus property.
- Instantiate a Manager and demonstrate property inheritance.

6. Create a base class Vehicle with a virtual method Drive().

1. Override Drive() in derived classes Car and Bike.
2. Demonstrate polymorphic behavior by calling Drive() on different objects using base class references.

7. Define an IShape interface with a method CalculateArea().

1. Implement the interface in Circle and Rectangle classes.
2. Instantiate objects and call CalculateArea() for different shapes

8. Create 2 classes Fulltime and Parttime with following properties and methods.

| Fulltime emp | Parttime Emp |
|---|---|
| Empid | Empid |
| Fname | Fname |
| Lname | Lname |
| Dept | Dept |
| Monthly salary | Totalhoursworked |
| | RatePerhour |

DisplayFullName()//logic to display fullname

DisplayAnnualSalary()//logic to calculate annual salary

Identity common properties from both the classes and move those to empbase class. Create empbase for same and inherit empbase to fulltime and parttime classes respectively

9. In Console Application create Employee Class Containing Following

properties and Methods

```
enum Deptlist{
Sales, HR
}
enum DesignationList
{
Manager , Employee
}
```

**Automatic properties**

```
public string Empid { get; set; }
public string EmpName { get; set; }
public Deptlist Dept { get; set; }
public DesignationList Designation { get; set; }
public int Salary { get; set; }
```

Methods

PrintEmployeeDetails()

Prints the details of employee

Validations

1. Empid should start with E
2. Dept should be either Sales or HR
3. Salary cannot be negative value
4. Designation should be Manager or Employee(Create Enum for same)

Display appropriate error messages

10. **Polymorphism with Virtual Methods**

- Create an abstract class Shape with an abstract method CalculateArea().
- Implement concrete classes like Circle, Rectangle, and Triangle.

11.Create an interface IPlayable with a method Play().

- Implement this interface in two classes, MusicPlayer and VideoPlayer.