```
using System;

namespace Oop;

public class Person{
    private string _name;
    public string Name{
        get { return _name; }
        set {
            if(value == null || value == "" || value.Length >=32)
            {
                throw new Exception("Invalid Name");
            }
            _name = value;
        }
    }
    private int _age;
    public int Age{
        get { return _age; }
        set {
            if(value <= 0 || value > 128)
            {
                throw new Exception("Invalid Age");
            }
            _age = value;
```

```
        }
    }

    public Person(string name , int age){
      Name = name;
      Age = age;
    }
    public virtual void Print(){
      Console.WriteLine($"My name is {Name}, my age is {Age}");
    }
}

public class Student : Person {
    private int _year;
    public int Year{
        get { return _year; }
        set {
            if(value < 1 || value > 5)
            {
            throw new Exception("Invalid Year");
            }
            _year = value;
        }
    }
    private float _gpa;
```

```
    public float Gpa{
        get { return _gpa; }
        set {
                if(value < 0 || value > 4)
                {
                throw new Exception("Invalid Gpa");
                }
                _gpa = value;
            }
    }

    public Student(string name , int age,  int year, float gpa) : base(name , age){
        Year = year;
        Gpa = gpa;
    }

    public override void Print(){
        Console.WriteLine($"My name is {Name}, my age is {Age}, and gpa is {Gpa}");
    }
}



public class Database{

    int _currentIndex;
```

```
    public Person[] People = new Person[50];

    public void AddStudent(Student student){
        People[_currentIndex++] = student;
    }

    public void AddStaff(Staff staff){
        People[_currentIndex++] = staff;
    }

    public void AddPerson(Person person){
        People[_currentIndex++] = person;
    }
    public void PrintAll(){
        foreach(var person in People){
            person?.Print();
        }
    }

}

public class Staff : Person {
    private double _salary;
    public double Salary{
```

```
    get { return _salary; }
    set {
          if(value < 0 || value > 120000)
          {
             throw new Exception("Invalid Salary");
          }
          _salary = value;
       }
}
private int _joinYear;
public int JoinYear{
    get { return _joinYear; }
    set {
          var compare = 2022 - (2022-Age);
          if(compare <= 21)
          {
             throw new Exception("Invalid JoinYear");
          }
          _joinYear = compare;
       }
}

public Staff(string name , int age, double salary, int joinYear) : base(name , age){
    Salary = salary;
    JoinYear = joinYear;
```

```
    }

    public override void Print(){
        Console.WriteLine($"My name is {Name}, my age is {Age}, and my salary is {Salary}");
    }

}

public class Task
{
    private static void Main()
    {
        var database = new Database();

        while(true){
            Console.WriteLine("1.(Student 2.(Staff 3.(Person 4.(Print All");
            Console.WriteLine("Option: ");
            var option = Convert.ToInt32(Console.ReadLine());

            switch(option){
            case 1:
                Console.Write("Name: ");
                var name = Console.ReadLine();
                Console.Write("Age: ");
                var age = Convert.ToInt32(Console.ReadLine());
```

```
        Console.Write("Year: ");
        var year = Convert.ToInt32(Console.ReadLine());
        Console.Write("Gpa: ");
        var gpa = Convert.ToSingle(Console.ReadLine());

        try{
            var student = new Student(name, age, year, gpa);
            database.AddStudent(student);
        }
        catch(Exception e)
        {
            Console.WriteLine(e.Message);
        }
        break;
    case 2:
        Console.Write("Name: ");
        var name2 = Console.ReadLine();
        Console.Write("Age: ");
        var age2 = Convert.ToInt32(Console.ReadLine());
        Console.Write("Salary: ");
        var salary = Convert.ToDouble(Console.ReadLine());
        Console.Write("JoinYear: ");
        var joinYear = Convert.ToInt32(Console.ReadLine());

        try{
```

```
        var staff = new Staff(name2, age2, salary, joinYear);
        database.AddStaff(staff);

    }
    catch(Exception e)
    {
        Console.WriteLine(e.Message);
    }
    break;
case 3:
    Console.Write("Name: ");
    var name3 = Console.ReadLine();
    Console.Write("Age: ");
    var age3 = Convert.ToInt32(Console.ReadLine());

    try{
        var person = new Person(name3, age3);
        database.AddPerson(person);
    }
    catch(Exception e)
    {
        Console.WriteLine(e.Message);
    }
    break;
case 4:
```

```
            database.PrintAll();
            break;
        default:
            return;
        }
    }
  }
};
```