Day 2 recap question

1.What is the output of the following code snippet?

class Employee{

public string employeeName;

public Employee(string empName){

this.employeeName=empName;

}

}

class Program{

public static void Main(string[] args){

Employee emp1=new Employee("Harris");

Employee emp2=new Employee();

}

}

2. What is the output of the following code snippet?

class Calculator

{

private static int num=10;

public static void Reset()

{

num=0;

}

public static void Add()

{

num+=10;

}

public static void Sub(int num)

{

num-=5;

}

public static void Show()

{

Console.Write(num+ " ");

}

}

public class Program

{

public static void Main(string[] args)

{

int num=55;

Calculator.Sub(num);

Console.Write(num+ " ");

Calculator .Add();

Console.Write(num+ " ");

Calculator .Show();

}

}

 3. what is anonymous functions? List out some Example

4. what is Encapsulation?

5. how to assign and read the value of private variable of the class?

6. write a program to set only positive integer for price variable(private) of the product class.

7. Describe about Abstract Class.

8. Predict the outcome of the following code snippet?

public abstract class A{

public A()

{

Console.WriteLine("Hello from class A");

}

public abstract void Display();

}

public class B:A{

public B()

{

Console.WriteLine("Hello from class B");

}

}

public class Program{

static void Main(String[] args)

{

B b = new B();

b.Display();

}

}

9. What is an interface and when can use it?

10. Write similarities of abstract class and interface?

11. What is Method overloading give an Example

12. What is dynamic polymorphism and give an example

13. Method hiding and give an example

14. I have class that should inherit by any other class. How to do so?

15. Can we restrict method should not be inherited. If yes reason and Example If No give a reason and example

16. is the below code works without any error?

interface IUser

{

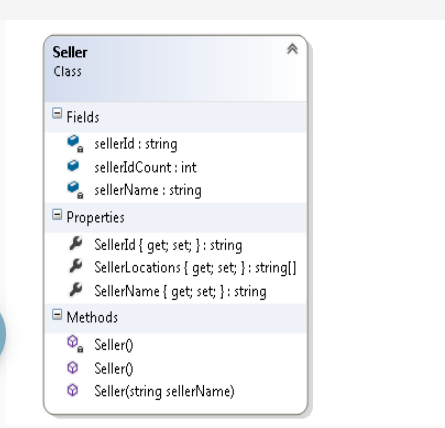
private string FirstName{get;set;}

private string LastName{get;set;}

void PrintDetails();

}

17. refer the below class diagram



**Seller Class**

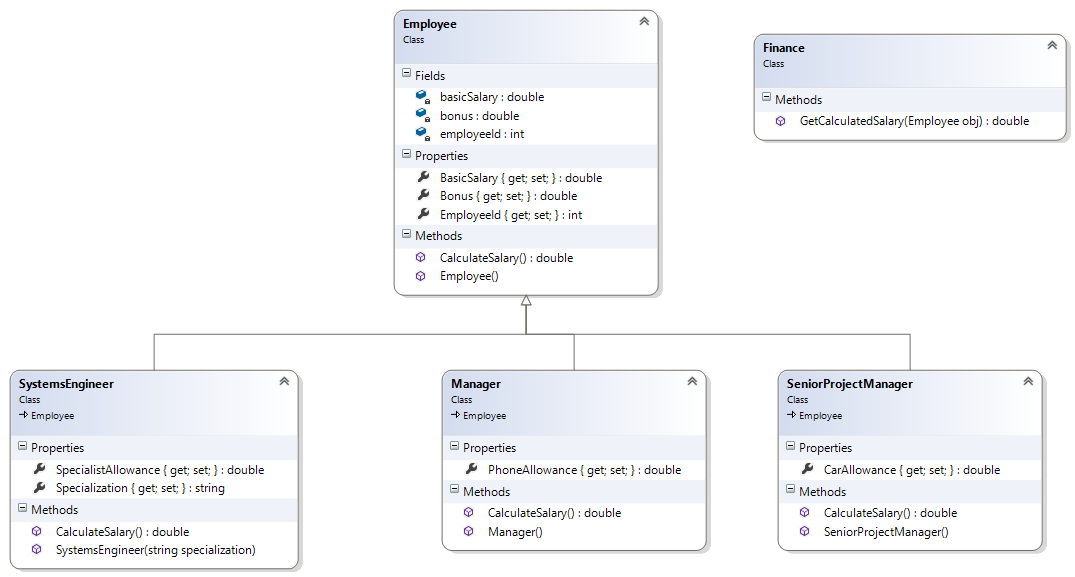
* Static constructor- Seller( )
  + The static variable ‘sellerIdCount’ should be initialized to 1001.
* Default constructor- Seller( )
  + The ‘sellerId’ should start with ‘S1001’ and should be incremented by 1 when a new object is created with the help of ‘sellerIdCount’. Eg: For the first seller the ‘SellerId’ should be S1001, for the second seller the ‘SellerId’ should be S1002 and so on.

## 18. **ProblemStatement:**

The Employee Management System wants to calculate the employee salary by setting appropriate values for allowances depending on the designation of the employee.

Refer the class diagram given below to implement the requirement.

**Class Diagram:**



* **Employee Class:**
  + **Default Constructor – Employee():**
    - **Initialize the value for BasicSalary as INR 10000/-.**
  + **CalculateSalary() method:**
    - **This is a virtual method which will be further overridden in the child classes. This method is expected to return the BasicSalary.**

* **SystemsEngineer Class**
  + **Specialization and SpecialistAllowance are Auto-Implemented properties.**
  + **Parameterized Constructor - SystemsEngineer (string specialization):**
    - **Invoke the Base Class constructor.**
    - **Initialize the Specialization Property to the value passed as argument.**
  + **CalculateSalary() method:**
    - **If Specialization is in any of the “C#” or “Java” or “SQL”, the SpecialistAllowance should be set to INR 3000/-.**
    - **For all other cases it should be set to 0.0**
    - **Salary is calculated as the sum of BasicSalary and SpecialistAllowance**
    - **Return Salary**

* **Manager Class**
  + **PhoneAllowance is an Auto-Implemented property.**
  + **Default Constructor – Manager():**
    - **Initialize the PhoneAllowance to INR 4000/-.**
  + **CalculateSalary() method:**
    - **Salary is calculated as the sum of BasicSalary and PhoneAllowance**
    - **Return Salary**

* **SeniorProjectManager Class:**
  + **CarAllowance is the Auto-Implemented property.**
  + **Default Constructor – SeniorProjectManager():**
    - **Initialize the CarAllowance to INR 6000/-.**
  + **CalculateSalary() method:**
    - **Salary is calculated as the sum of BasicSalary and CarAllowance**
    - **Return Salary**

* **Finance Class:**
  + **GetCalculatedSalary (Employee obj) method:**
    - **If the object passed is of type SystemsEngineer, the property Bonus is set to INR 5000/-**
    - **If the object passed is of type Manager, the property Bonus is set to INR 9000/-**
    - **If the object passed is of type SeniorProjectManager, the property Bonus is set to INR 15000/-**
    - **In all other cases Bonus should be 0.0**
    - **Invoke the method CalculateSalary() and add the returned value to Bonus.**
    - **Return the sum as the Total Salary**

**Note:**

* **Create an Instance of either of ‘SystemsEngineer’ or ‘Manager’ or ‘SeniorProjectManager’ with base class reference.**
* **Create an instance of Finance Class.**
* **Invoke the GetCalculatedSalary method, print the value returned and observe the output.**