Object Oriented Programming & String API

Q1:-

**Getters and Setters - Illustration**

Create a class named Product with the following private member variables.

* Id of type long
* ProductName of type string
* SupplierName of type string
* Include appropriate getters and setters.

2)create a display method print all values

Create another class and write a main method to test the above class.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output :**

**How many product**

**3**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

Q2:-

**Constructors : Illustration**

Create a class named Product with the following private member variables.

* Id of type long
* ProductName of type string
* SupplierName of type string

Include a 3-argument constructor and a default constructor.

Include a method named Display. It does not accept any arguments and its return type is void. Display the details of the product in this method. The method prototype is void Display();

Create another class and write a main method to test the above class.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output :**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

Q3:-

**Constructor Overloading - Illustration**

Create a class named Product with the following private member variables.

* Id of type long
* ProductName of type string
* SupplierName of type string

Include appropriate getters and setters.

Include a 3-argument constructor, 2 argument constructor and a default constructor. Assume that most of the products are supplied by “Nivas” suppliers and in the 2-argument constructor set the value of supplierName to “Nivas”.

Include a method named Display. It does not accept any arguments and its return type is void. Display the details of the product in this method. The method prototype is void Display();

Create another class and write a main method to test the above class.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output 1:**

Enter the product id

**1**

Enter the product name

**Printer**

Is the product supplied by Nivas Suppliers? Type yes or no (not case sensitive)

**Yes**

Product Id is 1

Product Name is Printer

Supplier Name is Nivas

**Sample Input and Output 2:**

Enter the product id

**1**

Enter the product name

**Printer**

Is the product supplied by Nivas Suppliers? Type yes or no (not case sensitive)

**NO**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

Q4:-

**ToString() and GetType() - Illustration**

Create a class named Product with the following private member variables.

* Id of type long
* ProductName of type string
* SupplierName of type string

Include a 3-argument constructor and a default constructor.

Override the ToString() method defined in the Object class. Display the details of the product in this method as shown in the sample output.

Create another class and write a main method to test the above class. Invoke the GetType() method from main.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output :**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

1 : Printer : HP

Invoking GetType() method : Product

Q5:-

**Equals() method – Illustration**

Create a class named Product with the following private member variables.

* Id of type long
* ProductName of type string
* SupplierName of type string

Include appropriate getters and setters.

Include a 3-argument constructor and a default constructor.

Include a method named Display. It does not accept any arguments and its return type is void. Display the details of the product in this method. The method prototype is void Display();

Create another class and write a main method to test the above class. In the main method, create 2 objects --- product1 and product2. Create a reference to the Product class product3 and make it refer to product2. Invoke the Equals() method using product1 and product2. Invoke the Equals method again using product2 and product3.

**Input and Output Format:**

Refer sample input and output for formatting specifications.

All text in bold corresponds to input and the rest corresponds to output.

**Sample Input and Output :**

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

Enter the product id

**1**

Enter the product name

**Printer**

Enter the supplier name

**HP**

Product Id is 1

Product Name is Printer

Supplier Name is HP

The two products are different

The two products are the same