**Lab Sheet - 02**

1. Create a C# console application that defines a class called Book with properties Title and Author. Instantiate an object of this class, set values for the properties, and display the information on the console.

using System;

class Book

{

public string Title { get; set; }

public string Author { get; set; }

}

class Program

{

static void Main(string[] args)

{

// Instantiate a Book object

Book myBook = new Book();

// Set values for properties

myBook.Title = "The Great Gatsby";

myBook.Author = "F. Scott Fitzgerald";

// Display information on the console

Console.WriteLine("Book Title: " + myBook.Title);

Console.WriteLine("Author: " + myBook.Author);

Console.ReadKey(); // Keep the console window open until a key is pressed

}

}

2. Develop a console program that models a simple bank account. Create a class named ‘BankAccount’ with properties AccountNumber and Balance. Implement a method ‘Deposit’ that allows the user to deposit money into the account. Instantiate an object, perform a deposit, and display the updated balance.

using System;

class BankAccount

{

public string AccountNumber { get; private set; }

public double Balance { get; private set; }

public BankAccount(string accountNumber)

{

AccountNumber = accountNumber;

Balance = 0;

}

public void Deposit(double amount)

{

Balance += amount;

}

}

class Program

{

static void Main(string[] args)

{

// Instantiate a BankAccount object

BankAccount account = new BankAccount("1234567890");

// Deposit money into the account

Console.Write("Enter the amount to deposit: ");

double depositAmount = double.Parse(Console.ReadLine());

account.Deposit(depositAmount);

// Display the updated balance

Console.WriteLine($"Updated Balance: ${account.Balance}");

Console.ReadLine(); // Keep the console open

}

}

3.Build an application to store and display the temperatures of a week using an array. Create a class named ‘TemperatureTracker’ with an array to store daily temperatures. Implement a method to display the temperatures. Instantiate an object, input temperatures, and display the weekly temperature report.

using System;

class TemperatureTracker

{

private double[] dailyTemperatures;

public TemperatureTracker(int daysInWeek)

{

dailyTemperatures = new double[daysInWeek];

}

public void InputTemperatures()

{

for (int i = 0; i < dailyTemperatures.Length; i++)

{

Console.Write($"Enter temperature for day {i + 1}: ");

dailyTemperatures[i] = double.Parse(Console.ReadLine());

}

}

public void DisplayWeeklyTemperatureReport()

{

Console.WriteLine("Weekly Temperature Report:");

for (int i = 0; i < dailyTemperatures.Length; i++)

{

Console.WriteLine($"Day {i + 1}: {dailyTemperatures[i]}°C");

}

}

}

class Program

{

static void Main(string[] args)

{

int daysInWeek = 7;

TemperatureTracker tracker = new TemperatureTracker(daysInWeek);

tracker.InputTemperatures();

Console.WriteLine();

tracker.DisplayWeeklyTemperatureReport();

Console.ReadLine(); // Keep the console open

}

}

1. Construct a C# program for a basic inventory system. Define a class named Product with properties ProductName and Price. Implement a parameterized constructor to initialize these properties. Instantiate objects using the constructor and display the product details.

using System;

class Product

{

public string ProductName { get; private set; }

public double Price { get; private set; }

public Product(string productName, double price)

{

ProductName = productName;

Price = price;

}

}

class Program

{

static void Main(string[] args)

{

// Instantiate objects using the constructor

Product product1 = new Product("Laptop", 999.99);

Product product2 = new Product("Smartphone", 699.99);

// Display product details

Console.WriteLine("Product 1:");

Console.WriteLine($"Product Name: {product1.ProductName}");

Console.WriteLine($"Price: ${product1.Price}");

Console.WriteLine();

Console.WriteLine("Product 2:");

Console.WriteLine($"Product Name: {product2.ProductName}");

Console.WriteLine($"Price: ${product2.Price}");

Console.ReadLine(); // Keep the console open

}

}

5.`Develop an application that simulates a library system. Create a class named LibraryBook with properties Title, Author, and Available. Implement a method BorrowBook that updates the availability status. Instantiate multiple book objects, perform book borrowing, and display the updated library status.

using System;

class LibraryBook

{

public string Title { get; private set; }

public string Author { get; private set; }

public bool Available { get; private set; }

public LibraryBook(string title, string author)

{

Title = title;

Author = author;

Available = true; // Initially, the book is available

}

public void BorrowBook()

{

if (Available)

{

Available = false;

Console.WriteLine($"The book '{Title}' by {Author} has been borrowed.");

}

else

{

Console.WriteLine($"Sorry, the book '{Title}' by {Author} is not available at the moment.");

}

}

}

class Program

{

static void Main(string[] args)

{

// Instantiate multiple book objects

LibraryBook book1 = new LibraryBook("To Kill a Mockingbird", "Harper Lee");

LibraryBook book2 = new LibraryBook("1984", "George Orwell");

LibraryBook book3 = new LibraryBook("The Great Gatsby", "F. Scott Fitzgerald");

// Perform book borrowing

book1.BorrowBook();

book2.BorrowBook();

book3.BorrowBook();

book1.BorrowBook(); // Attempt to borrow the same book again

// Display the updated library status

Console.WriteLine("\nLibrary Status:");

Console.WriteLine($"Book: '{book1.Title}' by {book1.Author}, Available: {book1.Available}");

Console.WriteLine($"Book: '{book2.Title}' by {book2.Author}, Available: {book2.Available}");

Console.WriteLine($"Book: '{book3.Title}' by {book3.Author}, Available: {book3.Available}");

Console.ReadLine(); // Keep the console open

}

}