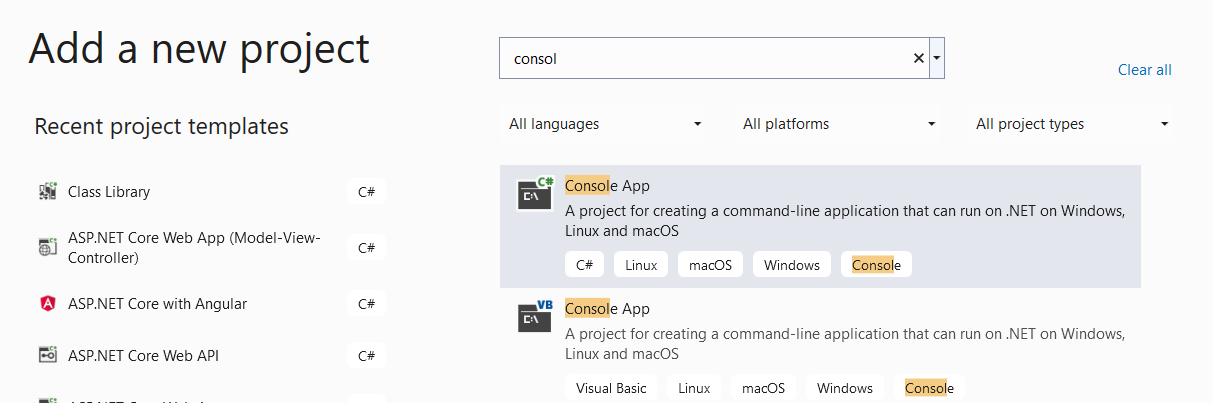
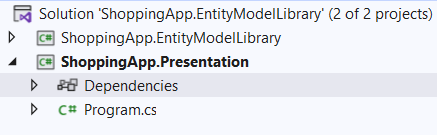
1. Open the ShoppingApp.EntityModelLibrary project created in the previous exercise
2. Right click on the solution in the Solution Explorer->Add-> New Project
3. Select console app

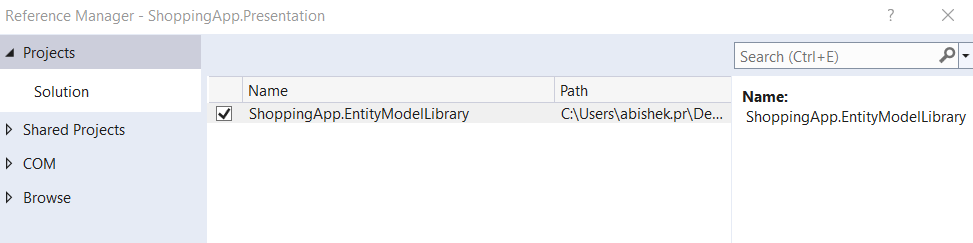


Name the app as ShoppingApp.Presentation

1. The Solution Explorer now displays 2 projects: the EF library and the Console App:



Right click on the Dependencies option under the Presentation App, Add Project Reference.



Select the ShoppingApp.EntityModelLibrary and add the reference

Now the console app will be able to access the entities and context of the library.

1. Modify the contents of the Program.cs file in the Console App as shown below:

public class Program

{

static ShoppingContext context;

static Program()

{

context = new ShoppingContext();

}

static void AddCategory(Category newCategory)

{

context.Categories.Add(newCategory);

context.SaveChanges();

}

static void ShowCategories()

{

foreach(Category category in context.Categories)

{

Console.WriteLine("{0} {1}",category.CategoryId,category.CategoryName);

}

}

static void UpdateCategory(Category updatedCategory)

{

Category category = context.Categories.Find(updatedCategory.CategoryId);

if(category != null)

{

category.CategoryName = updatedCategory.CategoryName;

context.SaveChanges();

}

}

static void DeleteCategory(int id)

{

Category category = context.Categories.Find(id);

if (category != null)

{

context.Categories.Remove(category);

context.SaveChanges();

}

}

static void Main(string[] args)

{

Console.WriteLine("testing CRUD operations");

Console.WriteLine("insert operation begins");

List<Category> categories = new List<Category>();

categories.Add(new Category { CategoryName = "Books" });

categories.Add(new Category { CategoryName = "Mobiles" });

foreach (Category c in categories)

{

AddCategory(c);

}

Console.WriteLine("insert operation ends");

Console.WriteLine("select operation begins");

ShowCategories();

Console.WriteLine("select operation ends");

Console.WriteLine("update operation begins");

Category category = new Category { CategoryId = 2, CategoryName = "Mobile Phones" };

UpdateCategory(category);

Console.WriteLine("update operation ends");

Console.WriteLine("showing updated rows");

ShowCategories();

Console.WriteLine("delete operation begins");

DeleteCategory(2);

Console.WriteLine("delete operation ends");

Console.WriteLine("showing rows after deletion");

ShowCategories();

}

}

1. Understand the code given in each method of the above program. Think of alternate approaches to perform update and delete operations.
2. Run the program and observe the output