using System;

class Program

{

static void Main(string[] args)

{

string referenceText = "Proverbs 3:5-6";

string scriptureText = "5. Trust in the Lord with all thine heart; and lean not unto thine own understanding. 6. In all thy ways acknowledge Him, and He shall direct thy paths.";

Scripture scripture = new Scripture(referenceText, scriptureText);

Reference reference = new Reference("Proverbs", 3, 5, 6);

Words words = new Words();

Console.WriteLine("Welcome to the Scripture Memorization Program!");

Console.WriteLine("Press Enter to start or type 'quit' to exit.");

string userInput = Console.ReadLine();

if (userInput != "quit")

{

Console.Clear();

scripture.Display();

while (!scripture.AllWordsHidden())

{

Console.WriteLine("Press Enter to continue or type 'quit' to exit.");

userInput = Console.ReadLine();

if (userInput == "quit")

break;

scripture.HideRandomWord();

Console.Clear();

scripture.Display();

}

Console.WriteLine("All words in the scripture are hidden.");

}

else

{

Console.WriteLine("Goodbye!");

}

}

}

public class Scripture

{

private string reference;

private List<string> words;

private List<bool> hiddenWords;

public Scripture(string reference, string text)

{

this.reference = reference;

words = new List<string>(text.Split(' '));

hiddenWords = new List<bool>(new bool[words.Count]);

}

public void Display()

{

Console.WriteLine(reference);

for (int i = 0; i < words.Count; i++)

{

if (hiddenWords[i])

{

Console.Write(new string('\_', words[i].Length) + " ");

}

else

{

Console.Write(words[i] + " ");

}

}

Console.WriteLine();

}

public bool AllWordsHidden()

{

return hiddenWords.TrueForAll(hidden => hidden);

}

public void HideRandomWord()

{

Random random = new Random();

int indexToHide = random.Next(0, words.Count);

if (!hiddenWords[indexToHide])

{

hiddenWords[indexToHide] = true;

}

}

}

using System;

public class Reference

{

private string \_book;

private int \_chapter;

private int \_startVerse;

private int \_endVerse;

public Reference(string book, int chapter, int startVerse)

{

\_book = book;

\_chapter = chapter;

\_startVerse = startVerse;

\_endVerse = 0;

}

public Reference(string book, int chapter, int startVerse, int endVerse)

{

\_book = book;

\_chapter = chapter;

\_startVerse = startVerse;

\_endVerse = endVerse;

}

public string GetSingleVerse()

{

return $"{\_book} {\_chapter}:{\_startVerse}";

}

public string GetMultiVerse()

{

if (\_endVerse > 0)

{

return $"{\_book} {\_chapter}:{\_startVerse}-{\_endVerse}";

}

else

{

return GetSingleVerse();

}

}

}

public class Words

{

private List<string> WordList;

private Random Random;

public Words()

{

WordList = new List<string>

{

"Trust", "in", "the", "Lord", "with", "all", "thine",

"heart;", "and", "lean", "not", "unto", "thine",

"own", "understanding.", "In", "all", "thy", "ways", "acknoledge",

"him,", "and", "he", "shall", "direct", "thy", "paths."

};

Random = new Random();

}

public string GetRandomWord()

{

int randomIndex = Random.Next(0, WordList.Count);

string randomWord = WordList[randomIndex];

WordList.RemoveAt(randomIndex);

return randomWord;

}

}