Question: Here are the instructions for a program in C++. What follows a···

| | Details |

Here are the instructions for a program in C++. What follows after that is a sample run. Thanks!

The idea of this program consists of writing a program that involves arrays, random number generation, and Strings. You may string objects.

- Write a program that applies the use of random-number generation to create sentences. Create four arrays with the following strings: (string objects highly suggested over c-strings) called `article`, `noun`, `verb`, and `preposition`.
- The arrays of Strings:
  - The article array should be const and contain the following articles: `"the"`, `"a"`, `"one"`, `"some"`, `"any"`
  - The preposition array should be const and contain the following prepositions: `"to"`, `"from"`, `"over"`, `"under"`, `"on"`
  - The noun array and the verb array should contain words entered by the user. So start the program by asking the user to enter 5 nouns (each separated by a space -- this should give you a hint as to how to read them in), and then ask them to enter 5 verbs (again, each separated by a space). Store these in the appropriate arrays.
  - Ensure that all the words stored in the noun array and the verb array are stored as all lowercase words. The user could enter a word with any case they choose, but it's up to you to store them as lowercase strings in your array. You may assume the user will enter valid verbs when prompted and valid nouns when prompted.
- Create a sentence by selecting a word at random from each array in the following order:
- `article, noun, verb, preposition, article, noun`

  This means, you need to generate a random value for each array to grab a word. For example: You cannot generate one random value (say, 1) and then take index 1 of each array. You must generate a random value to use as the index in each array you'll get a word from.
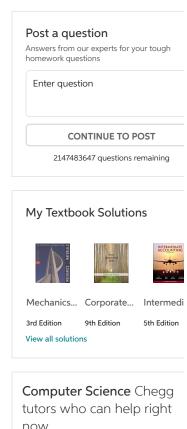
- As each word is picked, concatenate it to the previous words in the sentence. The words should be separated by spaces. When the final sentence is output, it should start with a capital letter and end with a period. Your program should generate random 20 sentences and output them to the console.
- Note: It's not enough to output just one word at a time. You need to create each sentence as a single string before printing it to output.
- Hint: The hardest part is making sure the sentence is capitalized, because you only have one article array (and it stores the lowercase version, since you use articles again). You may want to consider these ideas:
  - In the cctype library, there are functions that check for uppercase and lowercase
  - `At`, [] notation, and the + operation will also be helpful.
  - Please **don't** use global variables, and you can use any of the following libraries: iostream, iomanip, cctype, string, cstring, cstlib, and ctime.
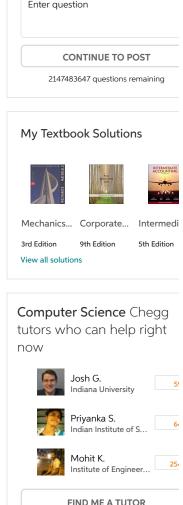
Sample Run 1

(user input is underlined)

```
Enter 5 nouns (lowercase): truck CAR dog MAN boy
Enter 5 past tense verbs (lowercase): ran jumped killed attacked cried

Some truck cried from the truck.
One boy jumped under the car.
Any car jumped from some man.
A car ran from one truck.
A truck attacked on a dog.
The man attacked on some boy.
One boy killed over any car.
A truck killed under the dog.
A car killed from a man.
Some dog attacked on the truck.
The man killed on one boy.
One car ran on the man.
The truck attacked from the dog.
A man killed on any man.
Any dog attacked over one truck.
One car ran on one boy.
A car killed under any car.
Some man attacked on a boy.
The boy cried under any dog.
The man jumped from one dog.
```

Chegg Study　　　TEXTBOOK SOLUTIONS　　EXPERT Q&A　　　Search

Anonymous answered this
1,365 answers

Recommend this answer　1　　0
Delete

```
#include<iostream>
#include<ctype.h>
#include<stdio.h>
#include<stdlib.h>
using namespace std;
//To accept 5 nouns and 5 verbs
void accept(string noun[5], string verb[5])
{
string n;
string v;
int t;
char ch;
cout<<"\n Enter 5 nouns: ";
//Loops 5 times to accept noun
for(int c = 0; c < 5; c++)
{
t = 0;
//Accept a noun
cin>>n;
//Loops till end of entered noun
while (n[t])
{
//Extracts a character
ch = n[t];
//Checks if it is upper case then converts it to lower case
if(isupper(ch))
n[t] = tolower(ch);
//increase the counter for entered noun index (n)
t++;
}//End of while loop
//Store the converted noun in the array of noun
noun[c] = n;
}//End of for loop
//Flushes the standard input
fflush(stdin);
cout<<"\n Enter 5 verbs: ";
//Loops 5 times to accept verb
for(int c = 0; c < 5; c++)
{
t = 0;
//Accept a verb
cin>>v;
//Loops till end of entered verb
while (v[t])
{
//Extracts a character
ch = v[t];
//Checks if it is upper case then converts it to lower case
if(isupper(ch))
v[t] = tolower(ch);
//increase the counter for entered verb index (c)
t++;
}//End of while loop
//Store the converted verb in the array of verb
verb[c] = v;
}//End of for loop
}//End of function
//To generate sentence
void generateSentence(string article[], string noun[], string verb[], string preposition[])
{
string sentence = "";
for(int x = 0; x < 20; x++)
{
//Generates 6 random numbers between 0 and 4
int f1 = rand() % 4;
int f2 = rand() % 4;
int f3 = rand() % 4;
int f4 = rand() % 4;
int f5 = rand() % 4;
int f6 = rand() % 4;
//Converts the first character to upper case
char ss = (char)toupper(article[f1][0]);
//Concatenates the random position of the arrays to form a sentence
sentence += ss + article[f1].substr(1) + " " + noun[f2] + " " + verb[f3] + " " + preposition[f4] + " " + article[f5] + "
```

```
//Re initializes the sentence to null for next sentence
sentence = "";
}//End of loop
}//End of function
//Main function
int main()
{
//Creates string type array
string article[] = {"the", "a", "one", "some", "any"};
string preposition[] = {"to", "from", "over", "under", "on"};
string noun[5];
string verb[5];
//Accept noun and verb
accept(noun, verb);
//Generates sentence
generateSentence(article, noun, verb, preposition);
}//End of main
```

Output:

Enter 5 nouns: truck CAR dog MAN boy

Enter 5 verbs: ran jumped Killed Attacked cried
A man killed to a truck.
One dog killed to a car.
A man jumped under some dog.
Some truck attacked to one car.
The dog jumped to one man.
Some dog attacked over a truck.
Some man attacked over some man.
One car jumped to a man.
A truck attacked to one car.
A man attacked from a dog.
The man killed over the dog.
The dog ran to one car.
One car killed over one car.
A truck jumped under the dog.
The truck killed to some truck.
The man killed under a dog.
One dog jumped from a man.
Some dog ran over a dog.
A dog jumped from the truck.
One car jumped from a truck.

Comment ❯

Your answer

Add your answer

Questions viewed by other students

Q: Here are the instructions for a program in C++. What follows after that is a sample run. If you're using functions, please include the prototype above the main function and the function definitions under the main function. The instructions should be clearly followed. Thanks! Sample run: Thanks!

A: See answer   100% (1 rating)

Q: Here are the instructions for a program in C++. What follows after that is a sample run. If you're using functions, please include the prototype above the main function and the function definitions under the main function. The instructions should be clearly followed. Thanks!

A: See answer   100% (1 rating)

Show more ⌄

Chegg Study

TEXTBOOK SOLUTIONS   **EXPERT Q&A**

Search

ABOUT CHEGG

Media Center
College Marketing
US Privacy Policy
Your CA Privacy Rights
International Privacy Policy
Terms of Use
General Policies
Cookie Notice
Intellectual Property Rights
Investor Relations
Corporate Development
Enrollment Services

RESOURCES

Site Map
Mobile
Publishers
Join Our Affiliate Program
Advertising Choices

TEXTBOOK LINKS

Return Your Books
Textbook Rental
eTextbooks
Used Textbooks
Cheap Textbooks
College Textbooks
Sell Textbooks

STUDENT SERVICES

Chegg Play
Study 101
Chegg Coupon
Scholarships
Career Search
Internships
College Search
College Majors
Scholarship Redemption

COMPANY

Jobs
Customer Service
Give Us Feedback
Chegg For Good
Become a Tutor

LEARNING SERVICES

Online Tutoring
Chegg Study Help
Solutions Manual
Tutors by City
GPA Calculator
Test Prep

PayPal VERIFIED   BBB   OVER 6 MILLION TREES PLANTED