Boys' High School & College

ISC – Class XII

String Manipulation

Question 1.
A class Rearrange has been defined to modify a word by bringing all the vowels in the word at the beginning followed by the consonants.
Example:
ORIGINAL becomes OIIARGNL
Some of the members of the class are given below:
Class name: Rearrange
Data Member/instance variable:
wrd: to store a word
newwrd: to store the rearranged word
Member functions/methods:
Rearrange(): default constructor
void readword(): to accept the word in UPPER case
vow freq_vow_con(): finds the frequency of vowels and consonants in the word and displays them with an appropriate message
void arrange(): rearranges the word by bringing the vowels at the beginning followed by consonants
void display(): displays the original word along with the rearranged word
Specify the class Rearrange, giving the details of the constructor(), void readword(), void freq _vow_con(), void arrange() and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

Question 2.
A class Capital has been defined to check whether a sentence has words beginning with a capital letter or not.
Some of the members of the class are given below:
Class name: Capital
Data member/instance variable:
sent: to store a sentence
freq: stores the frequency of words beginning with a capital letter
Member functions/methods:
Capital () : default constructor
void input (): to accept the sentence
boolean isCap(String w): checks and returns true if the word begins with a capital letter, otherwise returns false
void display(): displays the sentence along with the frequency of the words beginning with a capital letter
Specify the class Capital, giving the details of the constructor( ), void input( ), boolean isCap(String) and void display( ). Define the main( ) function to create an object and call the functions accordingly to enable the task.

Question 3.
A class SwapSort has been defined to perform string related operations on a word input.
Some of the members of the class are as follows:
Class name: SwapSort
Data members/instance variables:
wrd: to store a word
len: integer to store the length of the word
swapwrd: to store the swapped word
sortwrd: to store the sorted word
Member functions/methods:
SwapSort(): default constructor to initialize data members with legal initial values
void readword(): to accept a word in UPPER CASE
void swapchar(): to interchange/swap the first and last characters of the word in 'wrd' and stores the new word in 'swapwrd'
void sortword(): sorts the characters of the original word in alphabetical order and stores it in 'sortwrd'
void display(): displays the original word, swapped word and the sorted word
Specify the class SwapSort, giving the details of the constructor(), void readword(), void swapchar(), void sortword() and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

Question 4.
A class ConsChange has been defined with the following details:
Class name: ConsChange
Data members/instance variables:
word: stores the word len: stores the length of the word
Member functions/methods:
ConsChange(): default constructor
void readword(): accepts the word in lowercase
void shiftcons(): shifts all the consonants of the word at the beginning followed by the vowels (e.g. spoon becomes spnoo)
void changeword(): changes the case of all occurring consonants of the shifted word to uppercase, for e.g. (spnoo becomes SPNoo)
void show(): displays the original word, shifted word and the changed word
Specify the class ConsChange giving the details of the constructor), void readword ( ), void shiftcons (), void changeword() and void show(). Define the main() function to create an object and call the functions accordingly to enable the task.

Question 5.
A class TheString accepts a string of a maximum of 100 characters with only one blank space between the words.
Some of the members of the class are as follows:
Class name: TheString
Data members/instance variables:
str: to store a string
len: integer to store the length of the string
wordCount: integer to store the number of words
cons: integer to store the number of consonants

Member functions/methods:
TheString(): default constructor to initialize the data members
TheString(String ds): parameterized constructor to assign str=ds
void countFreq(): to count the number of words and the number of consonants and store them in wordCount and cons respectively
void display(): to display the original string, along with the number of words and the number of consonants
Specify the class TheString giving the details of the constructors, void countFreq() and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

Question 6.
A sequence of Fibonacci strings is generated as follows:
S0 = “a”, SF = “b”, Sn = S(n-1) + S(n-2) where ‘+’ denotes concatenation. Thus the sequence is:
a, b, ba, bab, babba, babbabab,……. n terms.
Design a class FiboString to generate Fibonacci strings. Some of the members of the class are given below:
Class name: FiboString
Data members/instance variables:
x: to store the first string
y: to store the second string
z: to store the concatenation of the previous two strings
n: to store the number of terms
Member functions/methods:
FiboString(): constructor to assign x=”a”, y=”b” and z=”ba”
void accept(): to accept the number of terms ‘n’
void generate(): to generate and print the Fibonacci strings. The sum of (‘+’ ie concatenation) first two strings is the third string. Eg. “a” is first string, “b” is second string then the third will be “ba”, and fourth will be “bab” and so on.
Specify the class FiboString, giving details of the constructor(), void accept() and void generate(). Define the main() function to create an object and call the functions accordingly to enable the task.

Question 7
Design a class Exchange to accept a sentence and interchange the first alphabet with the last alphabet for each word in the sentence, with single-letter word remaining unchanged. The words in the input sentence are separated by a single blank space and terminated by a full stop.
Example:
Input: It is a warm day.
Output: tI si a mraw yad
Some of the data members and member functions are given below:
Class name: Exchange
Data members/instance variables:
sent: stores the sentence
rev: to store the new sentence
size: stores the length of the sentence

Member functions:
Exchange(): default constructor
void readsentence(): to accept the sentence
void exfirstlast(): extract each word and interchange the first and last alphabet of the word and form a new sentence rev using the changed words
void display(): display the original sentence along with the new changed sentence.
Specify the class Exchange giving details of the constructor ( ), void readsentence (), void exfirstlast () and void display (). Define the main () function to create an object and call the functions accordingly to enable the task.