Imperative and Object Oriented Programming

Tutorial 4

Charlie Abela
Department of Artificial Intelligence
charlie.abela@um.edu.mt
Tutorial 4

In this tutorial you will be looking at exercises related to the topics of arrays and strings

Problems

1. Write a program consisting of two classes, a BitString class and its Launcher class. The former allows the user to enter a string of bits and a pattern that needs to be removed. The returned result is a sub-string with the remaining bits.

2. Write a program, RemoveVowel which allows the user to enter a phrase and a vowel, and then remove all occurrences of the vowel and return the stripped phrase.

3. Write a program, ReplaceVowels, which allows the user to enter a phrase and it replaces all vowel occurrences with an X.

4. Write a program, PatternSearch, which given a bit pattern and matching pattern, counts the number of times that the matching pattern is found in the original pattern.

5. Write a program called ProcessPersonArray which allows the user to:

   create a number of Person objects
   search for a Person by name
   find the oldest and youngest person by age
   find the average age of a number of Persons

   Each Person has an age, gender and name, together with relevant set/get methods. Use a Launcher class.

6. Given the following set of payscales:

   {10.50, 12.00, 14.50, 16.75, 18.00},
   {20.50, 22.25, 24.00, 26.25, 28.00},
   {34.00, 36.50, 38.00, 40.35, 43.00};

   the program called, PayScale, should allow the users to:
   i. find the average pay for a row
ii. find the difference between lowest and highest wages in each
payscale level
iii. prints the table values
iv. increases values in table by an entered amount

7. Write a program called, CaesarCipher, which implements the Caesar cipher
encryption algorithm. The algorithm essentially works as follows: each
character in the original string is shifted N positions, for e.g. if N = 1 and
the message was “I drink only decaf”, then the encrypted message would be
“J ! esjol ! pomz ! efdbg”. The program should allow the user to enter a
message and the value of N and displays the encrypted message. Use a
“wrap around” technique so that if the character to encrypt is “z” then the
program should go back to use a character from the start of the character
list. The program should also allow the user to decrypt the message back
to its original form.

8. Another encryption algorithm is called the Vignere cipher. This is similar to
the Caesar cipher; however a key (e.g. composed of capital letters) is
cyclically applied to the original message. So for example:

```
I drink only decaf
+++++++ . . . +
COFFEECOFFEECOFFEE
K-iW . . . . j
```

The values of the key’s characters are added to the values of the characters
in the original message. The values in the key could be assigned as
follows: 0 for A, 1 for B, 2 for C etc. So in the example above, I + 2 gives K.
Decryption reverses the process to generate the original message.

9. Write a complete address book maintenance application. The user of the
program has four options: add a new person (name, age, date of birth
address and email), delete a person, modify data of a person and search
for a person by a given name.

10. Write a program that allows the user to store a set of numbers in various
data structures: List, Stack and Queue. The user is prompted, initially to
select one of the structures and then allowed to enter and store a defined
set of numbers in an appropriately created structure. A List is a linked list
of numbers, starting from the first entered number it is possible to traverse
the list to the next numbers. It is suggested that a Node class with value
and reference to the next element is used. A Stack is a last-in-first-out
structure, while a queue is a first-in-first-out type of structure.