SE 116 – LAB#1 2014 – 2015 SPRING

Aim: Getting familiar with programming in C++; sample codes to understand the main difference between pass-by-value and pass-by-reference; using dynamic memory allocation; practicing the use of strings.

1. Define the following function that takes a double as an argument and returns the square of that value.

```
double squareByValue( double ); // sample function prototype
```

Define another function that takes a double reference as an argument and calculates the original values' square through its reference. The function returns void.

```
void squareByReference( double & ); // sample function prototype
```

Implement a simple program to test and compare the above functions. Try to understand the main difference between pass-by-value (call-by-value) and pass-by-reference (call-by-reference).

2.a. Define a sample "main" function that reads the year user began to primary school and prints out for how many years the user has been studying as a student. (Assume that the user is a student now.) Use two "integer" variables, "year" and "duration", as to be allocated statically in the memory.

Sample Output:

Please enter the year that you began to primary school: 2002
You have been studying as a student for 13 years.

b. Modify your program as in the following way: Change the type of the corresponding "year" variable to "short integer". Do not use a direct variable for "duration"; use a "durationPointer" variable as a "pointer to an integer" which will be allocated dynamically in the memory.

3. Write a simple program to test the type "string" using an array of strings. In "main", define an array of 5 strings, read 5 words into the array, finally find and print out the longest string in the array.
Hint: length() returns the number of the characters in a string.

Sample output:

Enter 5 words:
Ram
Screen
Computer
Lecture
class

The longest word is Computer.