BSc (Hons) in Information Technology
Year 1

Lab Sheet 07

IT1010 – Introduction to Programming Semester 1, 2021

Objectives:
At the end of the class the students should be able to:

- Implement simple functions in C language
- Write assert statements to test user defined functions

Exercise 1

This is a sample C program that includes two functions.

findGrade() – To find and return the grade of a module when mark is given as the parameter of the function.
testGrade() – This function includes two assert statements to test above function.

The main program reads the mark of a module as a user input, calls testGrade() , findGrade() functions and displays the grade.

Assumption : The user will input valid mark in between 0 and 100.

```
#include <stdio.h>
#include <assert.h>
char findGrade(float mark); //Function declarations
void testGrade();
int main(void)
{
	testGrade();

	float mark;

	printf("Enter mark of the module : ");
	scanf("%f", &mark);

	printf("Grade = %c\n", findGrade(mark));

	return 0;
}

char findGrade(float mark) //Function implementation
{
	if(mark >= 75)
		return 'A';
	else if(mark >= 65)
		return 'B';
	else if(mark >= 45)
		return 'C';
	else
		return 'F';
}
```

```
void testGrade()
{
      assert(findGrade(85) == 'A');
      assert(findGrade(25) == 'F');

}
```

i). Type the given C program in Dev C++.
ii). Compile and run the C program.
iii). Set a break point at the first statement in the main program.
iv). Using debugging option, add watches to the declared variables.
v). Using next line button, execute remaining statements and check how variable values are changed when the function is called with different user input values.

Exercise 2

This is a sample C program which has a function called ***calMark(***) to calculate and return the overall mark of a module. The function takes the marks obtained for the final exam (out of 100) and the marks obtained for the continuous assessments (out of 100) as the parameters.

Students can obtain 40% from their continuous assessments and 60% from their final exam.

Mark 1 = final mark * (60 / 100.0)
Mark 2 = CA mark * (40 / 100.0)

The overall mark of a module can be calculated as follows.

overall Mark = Mark 1 + Mark 2

The main program reads the marks earned for the final exam (out of 100) and the marks earned for the continuous assessments (out of 100) as user inputs, calls calMark() function and displays the overall mark of a module.

Two assert statements to check the function also called in the main program.

```
#include <stdio.h>
#include <assert.h>
#include <math.h>
float calMark(float final, float CA); //Function declaration
int main(void)
{
      assert(fabs(calMark(100.0, 100.0) - 100.0) < 0.01);
      assert(fabs(calMark(60.0, 70.0) - 64.0) < 0.01);

      float markFinal, markCA;

      printf("Enter mark for final exam : ");
      scanf("%f", &markFinal);

      printf("Enter mark for continuous assessments : ");
      scanf("%f", &markCA);

       printf("Overall mark = %.2f\n", calMark(markFinal, markCA));

      return 0;
}


float calMark(float final, float CA) //Function implementation
{
      float mark;
      mark = final * 0.6 + CA * 0.4;
      return mark;
}
```

i). Type the given C program in Dev C++.

ii). Compile and run the C program.

iii). Set a break point at the first statement in the main program.

iv). Using debugging option, add watches to the declared variables.

v). Using next line button, execute remaining statements and check how variable values are changed when the function is called with different user input values.

Exercise 3

"Suwasetha" private hospital offers healthy living packages. The package types and the payments are mentioned bellow.

Type	Package Name	Payment
S	Stroke Prevention Package	15000/=
C	Cancer Screening Package	50000/=
H	Healthy Heart Check Package	30000/=

i). Write a function called ***calPayment()*** to calculate and return the payment of a patient when the package type has given as the parameter of the function.

ii). The hospital will give 10% of discount for all the senior citizens who get these healthy living packages. Write a function called ***calDiscount()*** to calculate and return the discount. The function should take the age of the patient and the payment of the patient as parameters. Patients whose age 60 years or above are considered as senior citizens.

iii). Write a function called ***testDiscount()*** which contains two assert statements to check the function implemented in part ii) above.

iv). In your main function,

a) Call testDiscount() function
b) Input the package type and the patient age from the keyboard.
c) Display the net amount needed to be paid using the above functions.

v). Compile and run the program.
vi). Set a break point at the first statement of the main program.
vii). Using debugging option, add watches to the declared variables.
viii). Using next line button, execute remaining statements and check how variable values are changed.