# WEEK 2 - CONTROL FLOW

You are required to do the following:

1. Lab Questions – please do the lab questions during the lab session.
2. Assignment Questions – please do the assignment questions online using the automated programming assignment submission system (APAS). You do not need to submit your code for grading for this assignment.

**Lab Tutor**: For this lab-tutorial session, please discuss about the solution for each question in the lab. You may allocate about 30 minutes for each question. No need to discuss about the assignment questions.

## Lab Questions

1. **(computeGrade)** Write a C program that prints the ID and grade of each student in a class. The input contains the student IDs and their marks. The range of the marks is from 0 to 100. The relationships of the marks and grades are given below:

| Grade | Mark |
|---|---|
| A | 100-75 |
| B | 74-65 |
| C | 64-55 |
| D | 54-45 |
| F | 44-0 |

Use the sentinel value –1 for student ID to indicate the end of user input.

Sample input and output sessions are given below:

(1) Test Case 1:

```
Enter Student ID:
11
Enter Mark:
56
Grade = C
Enter Student ID:
21
Enter Mark:
89
Grade = A
Enter Student ID:
31
Enter Mark:
34
Grade = F
Enter Student ID:
-1
```

(2) Test Case 2:

```
Enter Student ID:
-1
```

The code using if-else if-else statements is given below for your reference:

```
#include <stdio.h>
int main()
{
   int studentNumber = 0, mark;
   printf("Enter Student ID: \n");
   scanf("%d", &studentNumber);
   while (studentNumber != -1)
   {
      printf("Enter Mark: ");
      scanf("%d", &mark);
      if (mark >= 75)
         printf("Grade = A\n");
      else if (mark >= 65)
        printf("Grade = B\n");
      else if (mark >= 55)
         printf("Grade = C\n");
      else if (mark >= 45)
         printf("Grade = D\n");
      else
         printf("Grade = F\n");
      printf("Enter Student ID: ");
      scanf("%d", &studentNumber);
   }
   return 0;
}
```

Write the program using the **switch** statement.

A sample program template is given below.

```
#include <stdio.h>
int main()
{
   /* insert variable declarations here */

   printf("Enter Student ID: \n");
   scanf("%d", &studentNumber);
   while (studentNumber != -1)
   {
      /* Write your program code here */
   }
   return 0;
}
```

2. **(printAverage)** Write a C program that reads in several lines of non-negative integer numbers, computes the average for each line and prints out the average. The value –1 in each line of user input is used to indicate the end of input for that line.

Sample input and output sessions are given below:

(1) Test Case 1:

```
Enter number of lines:
2
Enter line 1 (end with -1):
2 4 6 8 -1
Average = 5.00
Enter line 2 (end with -1):
1 3 5 7 9 -1
Average = 5.00
```

(2) Test Case 2:

```
Enter number of lines:
1
Enter line 1 (end with -1):
1 2 3 4 -1
Average = 2.50
```

A sample program template is given below.

```
#include <stdio.h>
int main()
{
   int total, count, lines, input;
   double average;
   int i;

   printf("Enter number of lines: \n");
   scanf("%d", &lines);

   /* Write your program code here */

   return 0;
}
```

3. **(printPattern)** Write a C program that accepts a positive number *height* between 1 and 10 as its parameter value, and prints a triangular pattern according to *height*. Note that only 1, 2 and 3 are used to generate the patterns. A sample input and output session when the program is called is given below. For example, *pattern(2)* will print the following pattern:

```
1
22
333
```

while *pattern(7)* will print the following pattern:

```
1
22
333
1111
22222
333333
1111111
```

Sample input and output sessions are given below:

(1) Test Case 1:

```
Enter the height:
3
Pattern:
1
22
333
```

(2) Test Case 2:

```
Enter the height:
7
Pattern:
1
22
333
1111
22222
333333
1111111
```

A sample program template is given below.

```
#include <stdio.h>
int main()
{
   int row, col, height;
   int num = 0;
   printf("Enter the height: \n");
   scanf("%d", &height);
   printf("Pattern: \n");

   /* Write your program code here */

   return 0;
}
```

4. **(computeSeries)** Write a C program that computes the value of $e^X$ according to the following formula:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots + \frac{x^{10}}{10!}$$

Sample input and output sessions are given below:

(1) Test Case 1:

```
Enter x:
0.9
Result = 2.46
```

(2) Test Case 2:

```
Enter x:
0
Result = 1.00
```

(3) Test Case 3:

```
Enter x:
-0.9
Result = 0.41
```

A sample program template is given below.

```
#include <stdio.h>
int main()
{
   int n, denominator = 1;
   float x, result = 1.0, numerator = 1.0;

   printf("Enter x: \n");
   scanf("%f", &x);

   /* Write your program code here */

   printf("Result = %.2f\n",  result);
   return 0;
}
```

## Assignment Questions

You may submit your code to the Automated Programming Submission and Assessment System (APAS) for testing.

1. (**computeNetPay**) Write a C program that reads the hours an employee worked in a week, computes the gross pay and income tax, and prints the gross pay, income tax and net pay on the screen. Assume that the pay structure and tax rate are given as follows: (1) the basic pay rate is $6.00 per hour; (2) the over-time pay rate (in excess of 40 hours) is one and a half time of the

basic pay rate; and (3) the tax rate is 10\% of the first $1000 of the gross pay, 20\% of the next $500 and 30% of the rest.

Sample input and output sessions are given below:

(1) Test Case 1:

```
Enter hours of work:
37
Gross pay = 222.00
Tax = 22.20
Net pay = 199.80
```

(2) Test Case 2:

```
Enter hours of work:
50
Gross pay = 330.00
Tax = 33.00
Net pay = 297.00
```

A sample program is given below:

```
#include <stdio.h>
int main()
{
   int hours;
   float tax, grossPay, netPay;

   printf("Enter hours of work: \n");
   scanf("%d", &hours);

   /* Write your program code here */

   printf("Gross pay = %.2f\n", grossPay);
   printf("Tax = %.2f\n", tax);
   printf("Net pay = %.2f\n", netPay);
   return 0;
}
```

2. (**computeSalaryGrade**) The salary scheme for a company is given as follows:

   Salary range for grade A: $700 - $899
   Salary range for grade B: $600 - $799
   Salary range for grade C: $500 - $649

In addition, a person whose salary is between $600 and $649 is in grade C if his merit points are below 10, otherwise he is in grade B. A person whose salary is between $700 and $799 is in grade B if his merit points are below 20, otherwise he is in grade A. Write a program to read in a person's salary and his merit points, and displays his grade.

Sample input and output sessions are given below:

(1) Test Case 1:

```
Enter the salary:
700
Enter the merit:
20
The grade: A
```

(2) Test Case 2:

```
Enter the salary:
500
Enter the merit:
```

```
20
The grade: C
```

A sample program is given below:

```
#include <stdio.h>
int main()
{
   int salary, merit;

   printf("Enter the salary: \n");
   scanf("%d", &salary);
   printf("Enter the merit: \n");
   scanf("%d", &merit);

   /* Write your program code here */

   return 0;
}
```

3. (**computeCarPrice**) Write a program to calculate the actual cost of buying a car in Singapore. Your program should input the list price and the category of a car, and print out the actual cost. Based on the list price, assuming that the car dealer will give a discount of 10%. You should also consider that there is a 10% luxury tax on the amount over $100,000 (after the discount), and that the G.S.T. tax is 3%. All taxes are computed based on the discounted price. Certificate of Entitlement (COE) must be obtained for every car in Singapore. The amount you bid for the COE is based on the category your car belongs to. COE is not taxed. The categories and their COE prices are given below:

   (1) Car (1600 c.c. & below) & Taxi: COE = 70,000
   (2) Car (above 1600 c.c.): COE = 80,000
   (3) Goods Vehicle & Bus: COE = 23,000
   (4) Motorcycle - COE: 600

   Sample input and output sessions are given below:

   (1) Test Case 1:

```
Please enter the list price:
30000
Please enter the category:
1
Total price is $97810.00
```

   (2) Test Case 2:

```
Please enter the list price:
50000
Please enter the category:
2
Total price is $54350.00
```

   A sample program is given below:

```
#include <stdio.h>
int main()
{
   int list, coe = 0, cat;
   double discounted, luxury = 0, gst, total;

   printf("Please enter the list price: \n");
   scanf("%d", &list);
   printf("Please enter the category: \n");
   scanf("%d", &cat);
```

```
    /* Write your program code here */

    printf("Total price is $%.2lf\n", total);
    return 0;
}
```

4. (**countChars**) Write a C program that reads in character by character from an input source, until '#' is entered. The output of the program is the number of English letters and the number of digits that appear in the input.

   Some sample input and output sessions are given below:

   (1) Test Case 1:

   ```
   Enter your characters (# to end):
   happy 34567 fans#
   The number of digits: 5
   The number of letters: 9
   ```

   (2) Test Case 2:

   ```
   Enter your characters (# to end):
   1a2b3c#
   The number of digits: 3
   The number of letters: 3
   ```

   A sample program is given below:

   ```
   #include <stdio.h>
   int main()
   {
      int ccount = 0, dcount = 0;
      char ch;
      printf("Enter your characters ('#' to end): \n");

      /* Write your program code here */

      printf("The number of digits: %d\n", dcount);
      printf("The number of letters: %d\n", ccount);
      return 0;
   }
   ```