

The American University in Cairo
Computer Science & Engineering Department
CSCE 1001 – Fundamentals of Computer Science

Lab Session 11

Top-Down Design with Functions

Objectives:

- To learn about functions and how to use them to modularize programs.
- To introduce structure charts as a system documentation tool.
- To understand how control flows between functions.
- To learn how to pass information to functions using arguments.
- To learn how to return a value from a function.

Problems:

1. Draw a structure chart and write a complete **modular** C++ program that produces a table showing the temperature equivalents in degrees Fahrenheit and degrees in Celsius. Let the user enter the starting, ending, and step values. The program validates all user input to accept only a start value not less than 50 and not more than 200. The computed Celsius value should be rounded to the nearest integer. Use the formula:

$$\text{celsTemp} = 5.0/9.0 (\text{fahrenTemp} - 32.00)$$

For example: for start value = 50, end value = 100, and step value = 10, the table should look like the following:

=====	
Fahrenheit	Celsius

50	10
60	16
....	
100	82
=====	

2. Draw a structure chart and write a complete **modular** C++ program that produces a table showing the area and circumference of circle for a range of radius values from 1 to 10 with a step of 1. Area and circumference should be printed rounded to the nearest hundredth. Use the formulas:

$$\text{Area of circle} = \pi \times \text{radius}^2$$

$$\text{Circumference} = 2 \times \pi \times \text{radius}$$

3. Write a C++ function that takes a positive integer number and returns its factorial. Mathematically, factorial of an integer number **n** is referred to as **n!** and defined as follows:

$$\mathbf{n! = n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1} \quad \text{for } \mathbf{n > 1}, \text{ and}$$
$$\mathbf{0! = 1}$$

4. Write a modular C++ program to compute the following **sum**:

$$\mathbf{\text{sum} = 0! + 1! + 2! + 3! + 4! + 5! + 6!}$$

5. Write a C++ function that takes a big positive integer number and returns the sum of its digits.
6. Write a C++ function that takes numbers, the first is a big positive integer and the second is just one digit. The function returns the number of occurrence of the digit in the big number. For example, if the big number is **1023226** and the digit is **2**, the function returns **3**, and if the big number **370554** and the digit is **9**, the function returns **0**.

7. Show the output of the following program segments

```

void drawLine(char);
float computeArea(int);

void main ()
{ drawLine('$');
  cout << setw(15) << "Radius" << setw(15) << "Area" << endl;
  drawLine('_');

  for (int k=1; k <= 10; k = k + 2)
    cout << setw(15) << k << setw(15) << computeArea(k) << endl;
  drawLine('#');
}

float computeArea (int r)
{ const float pie = 3.14;
  return ( pie * r * r);
}

void drawLine(char ss)
{
  for (int c = 1; c <= 30; c++)
    cout << ss;
  cout << endl;
}

```

```

bool myfunc(int n);
int main ()
{   for (int k=2; k <= 30; k += 3)
    if (myfunc(k))
        cout << setw(4) << k;
    cout << endl;
}

bool myfunc(int n)
{
  bool flag = true;
  int c = 2;
  while ((c <= n/2) && (flag))
    if (n % c == 0)
      flag = false;
    else
      c++;
  return flag;
}

```

```

int myfunc(int n);
void main ()
{
  int s = 0;
  for (k=0; k <= 6; k = k +2)
    s += myfunc(k);
  cout << "my result is" << setw(6) << s << endl;
}

int myfunc(int n)
{
  int f = 1;
  for (m=2; m <= n; m += 1)
    f *= m;
  return f;
}

```

```

int myDigit( long );
void main ()
{
  long num = 7658902;
  cout << "The required digit is " << setw(2) << myDigit(num) << endl;
  system ("pause");
}

int myDigit( long n )
{ const int ten = 10;
  int d = 100;

  do
  {
    if ( n % ten > d )
      d = n % ten;
    n = n / ten;
  } while ( n != 0 );

  return d;
}

```

Instructions:

- The TA gives hints how to solve problems 1, 3 and 5 (20 minutes)
- The TA gives hints how to solve two of the tracing programs (10 minutes)
- The TA asks students to solve problem 2 to be **submitted in Lab**.
- The TA asks students to solve all other problems 2 to be **submitted**.
-