Future University in Egypt – Faculty of Computer & Information Technology

CSC 112 Programming I Dr. Khalil	Midterm Exam III	Spring 2016
Student Name: ……		**ID:** ……

Question 1 (20 points)

Show the output of each of the following program segments:

```
int AB [3][3] = { 5, 3, 9, 4, 22, 9, 2, 5, 7}, T[3], S;
 for  ( int row = 0; row < 3; row++)
      {
        S = 0;
        for ( int col = 0; col < 3; col++)
            if ( AB[row][col]  % 2 == 0 )
               S = S + AB[row][col];
            T[row] = S;
      }
 cout << "The Final Content: " << endl;
 for  ( int row = 0; row < 3; row++)
      {
        for ( int col = 0; col < 3; col++)
            cout << setw (4) << AB[row][col];
        cout << setw(4) << T[row] << endl;
      }
```

```
int x = 3, y = 11;
while (x < 5)
 {   y = y / x++;
 cout << setw(3) << x << setw(3) << y << endl; }
```

```
int A[4] = {5, 2, 8, 9}, B[4] = {10,15 ,3, 18};
for (int k = 0; k < 4; k++)
    B[k] = B[k] % A[k];

cout << "Final Result:" << endl;

for (int n = 0; n < 4; n++)
    if (B[n] != 0)
      cout << B[n] << endl;
```

```
int B[3] = {5, 6, 8};
int sum;
for ( int c = 0; c < 3; c++)
{   sum = 1;
   for ( int f = 2; f <= B[c] / 2; f++)
     if ( B[c] % f == 0 )
        sum += f;
   if ( B[c] != sum )
       cout << B[c] << endl; }
```

```
int x = 5.7, y = 5.5;
while (x == y)
      x = 2 * x;
cout << setw(3) << x << setw(3) << y << endl;
```

Question 2 (15 points)

Draw the Flow Chart and the final output of the following program:

```
int a = 4.7, b = 6.6, c = 8.5, t;
if ((a > b) && (a > c))
        t = a;
else
         if (b > c)
            t = b;
         else
            t = c;
cout << "The required value: " << endl;
cout << a + b + c – t << endl;
```

Flow Chart

Output

Question 3 (15 points + 5 Bonus)

Write only a C++ function that takes two inputs, the first is a big positive integer and the second is just one single digit. The function checks the big integer number and returns **true** if the single digit does exist in the big number, and **false** otherwise. For example, if the big number is **1023226** and the single digit is **2**, the function returns **true**, and if the big number is **370554** and the single digit is **9**, the function returns **false**. (**Bonus of 5 marks** for making the function running efficiently).

Question 4 (15 points)

Write **only the nested C++ loops** to produce the following output:

```
#  2  3  4  5  6  7  8  9
&  &  3  4  5  6  7  8  9
#  #  #  4  5  6  7  8  9
&  &  &  &  5  6  7  8  9
#  #  #  #  #  6  7  8  9
&  &  &  &  &  &  7  8  9
#  #  #  #  #  #  #  8  9
&  &  &  &  &  &  &  &  9
#  #  #  #  #  #  #  #  #
```

The Prgram

Question 5 (15 points)

A Perfect integer number is a positive integer number where the sum of its factors (including 1) is equal to the number itself (for example, 6 is the first perfect number because 1 + 2 + 3 (the factors of 6) = 6). The next perfect number is 28 because 28 = 1 + 2 + 4 + 7 + 14 where 1, 2, 4, 7, 14 are the factors of 28.

The following modular C++ program computes and displays the sum of all perfect integers in the range of 2 up to 5000. There are some missings (represented by dots) in the given program. Complete these missings such that the program could be compiled and run correctly.

The Program

```
# include <iostream>
#include <iomanip>
using namespace std;

…….

void main ()

{   int sum ……;

     for (…….……..)

                ……

                          sum ……..

     cout << “The Sum of all Pefect integers in the Range = “ << setw(5) << …….. << endl;

     system ("pause");

} // End of main function


……  perfect ( ……. )

{   int ……;

   for ( int c ……..……..)

         If ( ……..…….. )

              ……..

    return  ( ……..…… );

} // End of perfect function
```

Question 6 (20 points)

Design and write a **modular** C++ program using functions to generate a table showing the conversion from Celsius temperature values to Fahrenheit temperature values according to the following formula:

Fahrenheit = 9 /5 Celsius + 32.00

The program should read the start integer Celsius (startCel), the end integer Celsius (endCel) and the step value (stepCel). The program should validate the user input through enforcing the following conditions:

- The start integer Celsius should be greater than -10 and not more than 45.
- The difference between the end integer Celsius and the start integer Celsius should not be less than 10.
- The step value should be greater than 0 and not greater than the difference between the end integer Celsius and the start integer Celsius.

The computed Fahrenheit value should be printed rounded to the **nearest integer value** without showing the decimal point.

As an Example: Start Integer Celsius (**starCel**) = **0**, End Integer Celsius (**endCel**) = **30**, Step Value (**stepCel**) = **5**

```
-----------------------------------------------
Celsius                 Fahrenheit
-----------------------------------------------
      0                         32
      5                         41
     10                         50
      …                        …….

     30                         86
-----------------------------------------------
```

C++ Modular Program

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

You can complete your program on the back sheet