National University of Computer and Emerging Sciences, Lahore Campus

Lab No 2			
Course Name:	Programming Fundamentals	Course Code:	CS 188
Program:	BS(SE)	Semester:	Fall 2020
Duration:	2.5 hours	Total Points:	10 + 15 + 15 20 + 20 + 20
Lab Date:	Saturday, October 3, 2020	Weight	2 + 3%
Section:	SE-1A and SE-1B	Page(s):	

Instruction/Notes: **Cheating during the lab will result in negative marks**

Topics Covered: Computational problem solving and writing programs using C++
In this lab we will mainly perform two types of activities

1. Convert a given pseudocode/Flow chart to a **C++** program.
2. Plan and write a C++ program to solve a given problem

You might use any IDE available at your computer or use one of the online IDE available at http://cpp.sh/, https://www.codechef.com/ide, https://ideone.com/, https://www.onlinegdb.com/

Activity 1:

Problem No 1:
It is a known fact that sum of angles of a triangle is 180^{o} and for any three values that add to 180 one can construct a triangle with angles equal to the given values.

The pseudocode given below can be used determine if a triangle can be created with angles equal to the values specified by the user. Use the pseudocode to **create a C++ program that uses ask the user to specify three positive angles and then display a message stating if a triangle can be created using these angle or not.**

```
10    DECLARE A, B, C
20    INPUT A, B, C
30    IF ( A + B  + C  == 180) THEN
40          PRINT “A Triangle can be created using these angles”
50    ELSE
60          PRINT “A Triangle cannot be created using these angles”
70    END IF
```

Problem No 2:

In the **Gregorian calendar** every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100, but these centurial years are leap years if they are exactly divisible by 400. For example, the years 1700, 1800, and 1900 are not leap years, but the years 1600 and 2000 are. This logic can be easily converted into a program that can determine if a year is a leap year or otherwise.

The following pseudocode has been written for this purpose and you are required to write a program using C++ to do the same.

```
10      DECLARE Year
15      INPUT Year
20      IF ( Year MOD 400  == 0) THEN
25              PRINT "LEAP YEAR"
30      ELSE
35              IF ( Year MOD 100  == 0) THEN
40                      PRINT "NOT A LEAP YEAR"
45              ELSE
50                      IF ( Year MOD 4  == 0) THEN
55                              PRINT "LEAP YEAR"
60                      ELSE
65                              PRINT "NOT A LEAP YEAR"
70                      END IF
75              END IF
80      END IF
```

Problem No 3:

Different character of English language and digits from 0 to 9 are represented using numeric codes (i.e. numbers) within a digital computer. One way to represent these is using ASCII (**A**merican **S**tandard **C**odes for **I**nformation **I**nterchange) codes. Following table shows the range of ASCII codes for various characters

Characters	ASCII Values
A – Z	65 – 90
a – z	97 – 122
0 – 9	48 – 57
special symbols	0 - 47, 58 - 64, 91 - 96, 123 – 127

So codes 65-90 are used for capital letters, codes 97-122 are used to code small letters, codes 48 to 57 are used to code numeric digits and all other codes below 128 are used to encode special characters.

The following pseudocode can be used to determine the type (i.e. capital, small, digit or special) of character coded by a given number. Convert this pseudocode into a working C++ program.

```
10    DECLARE Code
15    INPUT Code
20    IF ( Code > 64 AND code < 91) THEN
25        PRINT "Upper Case Character"
30    ELSE
35        IF (Code > 96 AND code < 123) THEN
40            PRINT "Lower Case Character"
45        ELSE
50            IF (Code > 47 AND code < 58) THEN
55                PRINT "A Digit"
60            ELSE
65                IF ( Code < 128) THEN
70                    PRINT "Special Character"
75                ELSE
80                    PRINT "NOT AN ASCII Character"
85                END IF
90            END IF
95        END IF
100   END IF
```

Activity 2:

Problem No 1:

Write a program that takes height of a person as input in **feet** and **inches**. The program should then convert the height in centimeters and display it on screen.

Your program should

1. Input the length in feet and inches.
2. Convert the length into total inches.
3. Convert total inches into centimeters.
4. Output the height of person in inches as well as in centimeters.

Following fact might be useful for creating the programs.

1 Foot = 12 inches
1 Inch = 2.54 centimeters

Problem No 2:

A common task performed by a cash clerk working at a famous store to compute the amount of money to be returned to the customer. The cash clerk knows the **total bill** and **amount paid** by the user and he then computes the **amount to be returned** and number of **notes/coins** of each type to be given back to the customer.

In this problem your job is to write a program that will take as input the total bill and the amount rendered/paid by the customer. The program should then print the amount to be given back to the customer along with the quantity of each type of notes/coins.

The program must compute the minimum number of notes of each type to make the job of clerk easier.

Further, you must assume that the notes of Rs. 5000, 1000, 500, 100, 50, 20, and 10, and coins of amount Rs. 5, 2 and 1 are used in Pakistan.

For example, when user enter a number bill amount **425** and cash given by customer as Rs. **1000** as the two inputs the program must display the following information in an easy to read format on screen.

```
Amount:        575
               500:   1
               50:    1
               20:    1
               5:     1
```

The program must also display a warning message on screen if amount paid is less than the total bill

Problem No 3:

Approximate values of **sin** and **cos** of an angle **x in radians** can be computed using the following series representations

sin x = x - x^3/3! + x^5/5! - x^7/7! + x^9/9!

&&
cos x = 1 - x^2/2! + x^4/4! - x^6/6! + x^8/8!

Use these approximations to compute the values of **sin**, **cos** and **tan** of an angle **x specified in degrees** by the user.

Following table lists values of factorials of integers between 1 and 9

Number	**Factorial**	**Number**	**Factorial**
1	1	6	720
2	2	7	5040
3	6	8	40320
4	24	9	362880
4	120	10	3628800

Please remember that the user will only input the value of x in degrees whereas the approximation works for angles given in radians. We know that π radians = 180^o where π = 3.1415.

Moreover, you are not allowed to use loops or any built-in functions for computing power of a number or factorial.