

## **Programming Fundamentals**

### **Assignment**

**Weightage=10**

- Plagiarized tasks will be awarded zero marks.
- There will be deduction for late submission.
- Use meaningful variable names instead of a, b, c, x, y, z.
- Use proper indentation.
- Your programs should satisfy all the requirements mentioned in the description or discussed in lab.
- Grading will also be based on your lab performance, better to complete tasks in lab.

1. Using nested loops print the following patterns according to user input (no of rows). All these patterns must be done in a single program.

```
1
2 3
4 5 6
7 8 9 10
```

```
0
0 1
0 1 1 2
0 1 1 2 3
```

```
A
B B
C C C
D D D D
```

```

how many rows? 5
Pattren 1:
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

Pattren 2:
0
0 1
0 1 1
0 1 1 2
0 1 1 2 3

Pattren 3:
A
B B
C C C
D D D D
E E E E E

```

2. There are two subscription plans for an antivirus software. Yearly subscription costs 100\$, Monthly subscription costs \$15. Write a program that asks how many subscriptions for each plans were availed by users, then displays the amount of income generated from those subscriptions. Format your dollar amount in fixed-point notation, with two decimal places of precision.
  - The task should be done using two functions i-e monthlySubscriptionIncome() and yearlySubscriptionIncome().
  - Each of these two functions will have one parameter i-e noOfUsers and will return the income generated from those plans. The total income generated should be displayed in main.

- The price for yearly and monthly subscriptions should be defined as a global const variable in your program.

```
Enter the no of users availed Yearly Subscription: 10
Enter the no of users availed monthly Subscription: 5
Total Revenue Generated: 1075$
-----
```

3. Write a program that takes the date of birth in the format of DD-MM-YYY and tells the person his zodiac sign. Further you can output a short description too about that star sign (qualities, personality etc. Recommended but optional). Take it as a fun task.

Observe the attached screenshot in detail. Your program should validate all inputs and give output like the screenshot attached.

#### **STAR SIGN DATES:**

Aries (March 21-April 19)  
Taurus (April 20-May 20)  
Gemini (May 21-June 20)  
Cancer (June 21-July 22)  
Leo (July 23-August 22)  
Virgo (August 23-September 22)  
Libra (September 23-October 22)  
Scorpio (October 23-November 21)  
Sagittarius (November 22-December 21)  
Capricorn (December 22-January 19)  
Aquarius (January 20-February 18)  
Pisces (February 19-March 20)

```

Enter your date of birth in this format DD-MM-YYYY: 28-02-1999
Pisces
Wanna check zodiac sign again? Y/N Y

Enter your date of birth in this format DD-MM-YYYY: 1-1-2000
Capricorn
Wanna check zodiac sign again? Y/N Y

Enter your date of birth in this format DD-MM-YYYY: 32-04-2001
Invalid day for April
Wanna check zodiac sign again? Y/N Y

Enter your date of birth in this format DD-MM-YYYY: 11-13-1999
Invalid Month
Enter your date of birth in this format DD-MM-YYYY: 03-03-2004
Pisces
Wanna check zodiac sign again? Y/N N
-----

```

4. A container can store 55 boxes of chocolates, while a box of chocolates can hold 22 chocolates. Create program that asks the user to enter: The quantity of chocolates Next, the application displays: the quantity of containers and boxes needed to ship the chocolates. Keep in mind that each container must hold the required number of boxes, and each box must have the required quantity of chocolates. You can eliminate the last box of chocolates and output the number of extra chocolates if it has less chocolates in it than the quantity you requested. Similar to this, you can eliminate the last container if it has less boxes than the number you provided and output the remaining boxes instead.
  
5. Create an application that asks the user to enter the monthly budgeted amount. The user should then be prompted via a loop to enter each of their monthly spending and maintain a running total. The application should show how much the user is spending more or less than they had planned after the loop is finished.

