



**National University**



Of Computer & Emerging Sciences Faisalabad-Chiniot Campus

## **National University of Computer and Emerging Sciences**

**Department of Computer Sciences**



### **CL-118 Programming Fundamentals Project**

<b>Course Instructor</b>	Dr. Rabia Maqsood
<b>Instructor</b>	M. Hannan Farooq
<b>Semester</b>	Fall 2020



## Objectives:

- All concepts of programming fundamentals
- Control Structure (Selection)
- Control Structure (Repetitive/ Loops)
- Function
- Array (one-dimensional, Multi-dimensional)
- File Handling

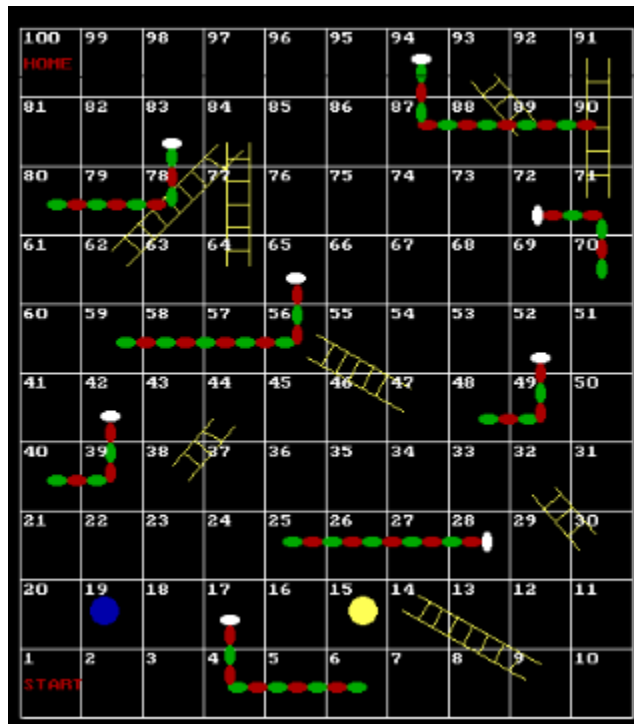
**Note:** Carefully read the following instructions (*Each instruction contains a weightage*)

1. Anyone found in copying the project from any other group or any other source, both the groups will be penalized with zero marks.
2. This game is an individual project.
3. There must be a block of comments at the start of the code, the block should contain a brief description of the functionality of the code. Also, comment with all functions about their functionality.
4. Proper indentation of code is essential.
5. Write a code in C++ language using Microsoft Visual Studio.
6. First, think about statement problems and then write/draw your logic on a copy.
7. After copy pencil work, code the problem statement on the Microsoft Studio C++ compiler.
8. At the end when you have done your project, submit a **complete project**
9. Submit your file in this format  
**20F-XXXX \_SnakeAndLadderGame.zip.**
10. Submit a **complete project file** with **your code copy-pasted on a word file with all possible output screenshots. Both project file and word file must be included in the .zip file.** Make your submission on Google Classroom (Make sure your submission is completed).

## **Grading Criteria**

1. Your program must be stable which means it should do what it is supposed to do. You can achieve these criteria by knowing adequate knowledge of how the game works and by carefully follow and implement all the rules mentioned below.
2. Implement your code with an efficient structure which means the use of control, repetition, and all other features we covered in the course, must be utilized efficiently.
3. Be careful while using functions signature (prototype) and try to avoid using break and continue statements. Use only very essentials global variable otherwise do not use a global variable. Do not use bad programming and hard code techniques.
4. Use proper indentation and comments where they are necessary (use with function definitions, function calls, checks, loops, variables, etc.)

## Problem: Snake and ladder game



## How to play game

You must use this link, how candy crush game works

<https://www.youtube.com/watch?v=a-KTZF2EEKc>

## Introduction

Snakes and Ladders, known originally as Moksha Patam, is an ancient Indian board game for two or more players regarded today as a worldwide classic. It is played on a game board with numbered, gridded squares. Several "ladders" and "snakes" are pictured on the board, each connecting two specific board squares. The object of the game is to navigate one's game piece, according to die rolls, from the start (bottom square) to the finish (Top Square), helped by climbing ladders but hindered by falling down snakes.

The size of the grid varies but is most commonly 8×8, 10×10 or 12×12 squares. Boards have snakes and ladders starting and ending on different squares; both factors affect the duration of play. Each player is represented by a distinct game piece token. A single die is rolled to

determine the random movement of a player's token in the traditional form of play; two dice may be used for a shorter game

## Features to implement (Rules for the game)

Following are features/ rules you must implement in your game in C++ language using all concepts you learn in the course.

The game is based on a menu that contains five options

### I. Play Game:

- As the game is based on four players so after selecting the play game option first game will get the names of all players by input the player's name and then your game will start from a toss, who won the toss has the first turn then the second, third and then the fourth player.
- The game contains a 10x10 board size that appeared on the console.
- Board contains at least five ladders to climb up and five snakes to bite down. Suppose if the player is on **S** (snake) index which is the 50<sup>th</sup> index on board then its updated position may be down to the 10<sup>th</sup> index and if the player is on **L** (ladder) index which is the 40<sup>th</sup> index on board then its updated position may be upon 70<sup>th</sup> index.

100	99	98	97	96	95	94	93	92	91
81	82	83	84	85	86	87	88	89	90
80	79	78	77	76	75	74	73	72	71
61	62	63	64	65	66	67	68	69	70
60	59	58	57	56	55	54	53	52	51
41	42	43	44	45	46	47	48	49	50
40	39	38	37	36	35	34	33	32	31
21	22	23	24	25	26	27	28	29	30
20	19	18	17	16	15	14	13	12	11
1	2	3	4	5	6	7	8	9	10

- Input must be randomly generated from 1 to 6.

- Any player can start playing by getting **six** on dice.
- If there's **six** on dice the player gets another turn to roll dice.
- If two or more players are at the **same point index** then all goes back to the initial state except the latest one.
- The first winner can give **six moves** forward to any other player as a gift.
- As soon as snakes bite a player screen color becomes **red** for few seconds and comes back to original color afterward.
- As soon as any player takes the ladder screen becomes **green** and comes back to the original color after few seconds.
- For this purpose, you need to use **windows.h** library and function **sleep(value)** use for the timer. You can use the **system ("cls")** for clear screen and **system("color 0-9 are for background and A-F are for foreground colors")** to change console color purposes. You may learn more about it using this link  
<https://www.sololearn.com/Discuss/552086/how-to-change-background-color-in-c>

II. **Credits:** This section contains who created the game.

III. **Instructions/ Rules:** How to play the game with all rules to play the game.

IV. **Record:** This section contains the name of the player who won the game with his/her number of turns to win game.

V. **Exit:** Your game simply exit.

Your game provides a choice to the user/player to select an option from 1 to 5 to proceed with the game.

Proper code indentation will hold extra marks!

Best of luck 😊

**You are done with your Project, submit it on Google Classroom at the given time.**