

### Quiz 1 Part A

Q1. Write short answers (2-3 lines) for the following questions:

- Considering two objects A and B, if a statement is written as A = B, any changes made to A or B afterwards will be reflected in both objects. Justify as True OR False.
- Why do we need copy constructors in C++?
- The compiler will always provide a default constructor for your class. Justify as True OR False.
- In C++, how can we ensure that dynamically created objects do not stay allocated in the memory after they are no longer being used?
- Which concept of OOP do the getters and setters signify?

Q2. Let's say you're working on a game called "Space Invaders". Multiple players can play this game, and the game itself stores the top three scores achieved by players. Create a class for your game with the following constraints:

- Following attributes and functionalities are available in your game:
  - A list of high scores
  - A list of usernames who scored these high scores
  - Username (of the current user playing the game)
  - Score (of the current user playing the game)
  - Constructors to initialize the initial score to 0. Default name should be "Player 1" if not specified by the user.
  - A function to display the high scores
  - A function to play the game
- Provide proper functionality for the following two functions:
  - void displayHighScore() – Should show a list of usernames and their high scores. Everyone should be able to see the highest scores obtained.
  - Void playGame() – Should allow the user to play the game – you don't have to provide functionality for gameplay – the function should generate a score (you can use the random function). This score is to be compared with the existing high scores. If the score is higher than any of the high scores, the ranking should be updated accordingly.  
For example:  
The current ranking is:  
1. Abeeha 9,999      2. Bakhtawer 8,888      3. Nida 7,777  
If the current player "XYZ" scores a 9,000 – The updated high score should be:  
1. Abeeha 9,999      2. XYZ 9,000      3. Bakhtawer 8,888

- Write a main function that will produce the following output by making use of your designed class, members, functions and objects:

1. Abeeha 9,999	2. XYZ 9,000	3. Bakhtawer 8,888
1. YourName 10,000	2. Abeeha 9,999	3. XYZ 9,000

## Quiz 1 Part B

Q1. Write short answers (2-3 lines) for the following questions:

- Considering two objects A and B, if a statement is written as  $A = B$ , any changes made to A or B afterwards will be reflected in both objects. Justify as True OR False.
- Why do we need copy constructors in C++?
- The compiler will always provide a default constructor for your class. Justify as True OR False.
- In C++, how can we ensure that dynamically created objects do not stay allocated in the memory after they are no longer being used?
- Which concept of OOP do the getters and setters signify?

## Question 2.

Imagine you're working on the backend development for "Galactic Conquest," a multiplayer space conquest game set in a vast galaxy. Each player aims to conquer different sectors of the galaxy and build their interstellar empire. The game tracks the top three conquerors based on their achievements in conquering sectors.

### Attributes and Functionalities:

List of Top Conquerors: Keeps track of the top three players who have conquered the most sectors.

List of Player Usernames: Records the usernames of players who conquered sectors.

Current Player Username: Represents the username of the current player playing the game.

Current Player's Conquests: Tracks the number of sectors conquered by the current player.

Constructors: Initializes the player's conquests to 0. Default username is "Player 1" if not specified.

Function to Display Top Conquerors: Shows the list of top conquerors along with the number of sectors conquered.

Function to Conquer Sectors: Simulates the conquest of sectors. Upon conquering a sector, the player's conquest count is incremented. If the player's conquest count surpasses any of the top conquerors, the rankings are updated accordingly.

### Functionality Details:

DisplayTopConquerors(): This function retrieves and displays the usernames and conquered sectors of the top three players in the game.

ConquerSectors(): When called, this function simulates the conquest of a sector by the current player. After conquering a sector, the player's conquest count is compared with the top conquerors' counts. If the player's count surpasses any of the top conquerors, their ranking is updated accordingly.

### Scenario:

In a typical session of Galactic Conquest, the game starts with an established list of top conquerors:

EmperorZorg: 20 sectors conquered

SpaceWarrior97: 18 sectors conquered

GalacticEmpress: 15 sectors conquered

As new players join or existing players conquer more sectors, the rankings dynamically change:

A new player named CosmicConqueror joins and conquers 17 sectors. The updated top conquerors list becomes:

EmperorZorg: 20 sectors conquered

CosmicConqueror: 17 sectors conquered

SpaceWarrior97: 18 sectors conquered

Later, GalacticEmpress conquers 19 more sectors, surpassing SpaceWarrior97. The updated top conquerors list becomes:

EmperorZorg: 20 sectors conquered

GalacticEmpress: 34 sectors conquered

CosmicConqueror: 17 sectors conquered