National University of Computer & Emerging Sciences, Karachi
Spring-2025 School of Computing
Quiz No. 1
13th February 2025

Course Code: CS-1004	Course Name: Object Oriented Programming
Instructor Name / Names: Ms. Atiya Jokhio	
Student-ID:	

Time Allowed: 30 minutes. **Total Points**: 12

TYPE A

Question:1 Suppose your program contains the following class definition (along with definitions of the member functions): **[6 points]**

```
class YourClass {
public:
YourClass(int new_info, char more_new_info);
YourClass();
void do_stuff();
private:
int information;
char more_information;
};
```

Which of the following are legal? If any statement is illegal, then write the correct (legal) statement.

1. YourClass an_object(42, 'A'); //legal
2. YourClass another_object; //legal
3. YourClass yet_another_object(); **// illegal YourClass yet_another_object;**
4. an_object = YourClass(99, 'B'); // legal
5. an_object = YourClass(); // legal
6. an_object = YourClass; **// illegal YourClass is a type, not an instance or function that returns an instance.**

Question:2 **[6 points]**

Design the class "PriateShip" with Attributes:

- Name: The name of the pirate ship ride (e.g., "The Black Pearl").
- Capacity: Maximum number of people the pirate ship can hold at a time.
- CurrentNumRiders: The number of people currently on the pirate ship.
- MaxSwingAngle: The maximum swing angle the pirate ship can achieve (in degrees).
- CurrentSwingAngle: The current swing angle of the pirate ship (in degrees).
- SwingInProgress: A boolean flag that indicates whether the swing is in progress.
- Design Parameterized Constructor which allows the user to set the name, capacity, and maximum swing angle, ensures capacity of at least 10 people (if less, default it to 10) and should not be in progress upon creation. And also, destructor for printing a message stating that the pirate ship ride is being destroyed.

Design a method named as " LoadShip" for adding a given number of riders to the pirate ship and If the number of riders exceeds available capacity, return the number of excess riders, another method named as " StartSwing " to start if it is not already in progress and if there are riders on board.

Note: create one object of the PirateShip class by using both the constructors.

```
#include <iostream>
#include <string>
```

```cpp
using namespace std;

class PirateShip {
private:
    string name;
    int capacity;
    int currentNumRiders;
    int maxSwingAngle;
    int currentSwingAngle;
    bool swingInProgress;

public:
    // Parameterized constructor
    PirateShip(string shipName, int cap, int maxAngle) {
        name = shipName;
        capacity = (cap >= 10) ? cap : 10;  // Ensure minimum capacity
        maxSwingAngle = maxAngle;
        currentNumRiders = 0;
        currentSwingAngle = 0;
        swingInProgress = false;
        cout << "Pirate Ship \"" << name << "\" created successfully.\n";
    }

    // Default constructor
    PirateShip() {
        name = "Unnamed Ship";
        capacity = 10;
        maxSwingAngle = 90;
        currentNumRiders = 0;
        currentSwingAngle = 0;
        swingInProgress = false;
        cout << "Default Pirate Ship \"" << name << "\" created.\n";
    }

    // Method to load riders
    int LoadShip(int riders) {
        int availableSpace = capacity - currentNumRiders;
        if (riders <= availableSpace) {
            currentNumRiders += riders;
             cout << riders << " riders loaded onto the ship \"" << name <<
"\".\n";
            return 0;
        } else {
            currentNumRiders = capacity;
            int excess = riders - availableSpace;
```

```
                cout << availableSpace << " riders loaded, " << excess << "
could not be accommodated.\n";
            return excess;
        }
    }

    // Method to start swing
    void StartSwing() {
        if (!swingInProgress && currentNumRiders > 0) {
            swingInProgress = true;
            currentSwingAngle = maxSwingAngle;
             cout << "The ship \"" << name << "\" has started swinging to "
<< maxSwingAngle << " degrees!\n";
        } else if (swingInProgress) {
            cout << "Swing is already in progress on \"" << name << "\".\n";
        } else {
             cout << "Cannot start swing: No riders on the ship \"" << name
<< "\".\n";
        }
    }

    // Destructor
    ~PirateShip() {
        cout << "Pirate Ship \"" << name << "\" is being destroyed.\n";
    }
};

// Main function to demonstrate functionality
int main() {
    // Create using parameterized constructor
    PirateShip blackPearl("The Black Pearl", 20, 120);
    blackPearl.LoadShip(15);
    blackPearl.StartSwing();

    cout << endl;

    // Create using default constructor
    PirateShip defaultShip;
    defaultShip.LoadShip(12);
    defaultShip.StartSwing();

    return 0;
}
```