# ASSESSMENT 1

## 1. Check if the entered username is valid without using a function to take input from the user

```cpp
#include <iostream>
#include <cctype>
using namespace std;

int main() {
    string username;
    cout << "Enter username: ";
    cin >> username;

    bool isValid = true;
    for (char c : username) {
        if (!isalnum(c)) {
            isValid = false;
            break;
        }
    }

    if (isValid) {
        cout << "The username is valid." << endl;
    } else {
        cout << "The username is invalid." << endl;
    }

    return 0;
}
```

**Output :**
Enter username: user123
The username is valid.

Enter username: user@123
The username is invalid.

## 2. Check if a given string is a palindrome using choice (with inline function)

```cpp
#include <iostream>
using namespace std;

inline bool isPalindrome(const string& str) {
    int n = str.length();
    for (int i = 0; i < n / 2; ++i) {
        if (str[i] != str[n - i - 1]) {
            return false;
        }
    }
    return true;
}

int main() {
    string str;
    cout << "Enter a string: ";
    cin >> str;

    if (isPalindrome(str)) {
        cout << "The string is a palindrome." << endl;
    } else {
        cout << "The string is not a palindrome." << endl;
    }

    return 0;
}
```

**Output :**
Enter a string: radar
The string is a palindrome.

Enter a string: hello
The string is not a palindrome.

## 3. Find and get square and cube using hierarchical inheritance

```cpp
#include <iostream>
using namespace std;

class Number {
protected:
    int num;
public:
    void setNumber(int n) {
        num = n;
    }
};

class Square : public Number {
public:
    int getSquare() {
        return num * num;
    }
};

class Cube : public Number {
public:
    int getCube() {
        return num * num * num;
    }
};

int main() {
    Square sq;
    Cube cu;
    int n;

    cout << "Enter a number: ";
    cin >> n;

    sq.setNumber(n);
    cu.setNumber(n);

    cout << "Square of " << n << " is: " << sq.getSquare() << endl;
    cout << "Cube of " << n << " is: " << cu.getCube() << endl;
```

```
    return 0;
}
```

**Output :**
Enter a number: 3
Square of 3 is: 9
Cube of 3 is: 27

## 4. Create a class named Rectangle with data members `length` and `breadth`, a function to calculate the area, and a constructor

```
#include <iostream>
using namespace std;

class Rectangle {
private:
    int length;
    int breadth;
public:
    Rectangle() {
        length = 0;
        breadth = 0;
    }

    Rectangle(int l, int b) {
        length = l;
        breadth = b;
    }

    int area() {
        return length * breadth;
    }
};

int main() {
    Rectangle rect1;
    Rectangle rect2(10, 5);

    cout << "Area of rectangle with default constructor: " << rect1.area() << endl;
```

```
    cout << "Area of rectangle with parameterized constructor: " << rect2.area() << endl;

    return 0;
}
```

**Output :**
Area of rectangle with default constructor: 0
Area of rectangle with parameterized constructor: 50