

OBJECT ORIENTED PROGRAMMING

LAB ASSIGNMENT -1

1. Write a program (WAP) to display "Hello World" on console display.

CODE:

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     cout<<"Hello world";
5     return 0;
6 }
```

OUTPUT:

```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/" && g++ Q1.C -o Q1
&& "/Users/yakshgupta/Desktop
/CPP/ASSIGNMENT1/"Q1
Hello world
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % █
```

2. Write a C++ program that will ask for a temperature in Celsius and display it in degree Fahrenheit.[F=9C/5+32]

CODE:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     float cels, fahren;
6     cout << "Enter temperature in Celsius: ";
7     cin >> cels;
8
9
10    fahren = (9.0/5.0) * cels + 32;
11    cout << "Temperature in Fahrenheit: " << fahren << "°F";
12
13    return 0;
14 }
```

OUTPUT:

```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/" && g++ Q2.C -o Q2
&& "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/"Q2
Enter temperature in Celsius: 100
Temperature in Fahrenheit: 212°F
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 %
```

3. WAP to demonstrate for, while, do-while (with all possible variations), like for loop can be demonstrated without giving initialization in for construct or without giving increment in for construct.

Sample:

```
for (int i=0; i<10; i++)
```

i=0

```
for (; i<10; i++)
```

i=0

```
for (; i<10;)
```

```
i++
```

CODE:

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5
6     cout << "Standard for loop:" << endl;
7     for (int i = 0; i < 10; i++) {
8         cout << "i = " << i << endl;
9     }
10    cout << endl;
11
12    int i = 0;
13    cout << "For loop without initialization:" << endl;
14    for (; i < 10; i++) {
15        cout << "i = " << i << endl;
16    }
17    cout << endl;
18
19    i = 0;
20    cout << "For loop without increment:" << endl;
21    for (; i < 10;) {
22        cout << "i = " << i << endl;
23        i++;
24    }
25    cout << endl;
26
27    i = 0;
28    cout << "For loop without initialization or increment:" << endl;
29    for (; i < 10;) {
```

```
29     for ( ; i < 10; ) {
30         cout << "i = " << i << endl;
31         i++;
32     }
33     cout << endl;
34
35     cout << "Standard while loop:" << endl;
36     i = 0;
37     while (i < 10) {
38         cout << "i = " << i << endl;
39         i++;
40     }
41     cout << endl;
42
43     i = 0;
44     cout << "While loop without increment:" << endl;
45     while (i < 10) {
46         cout << "i = " << i << endl;
47         i++;
48     }
49     cout << endl;
50
51     cout << "Standard do-while loop:" << endl;
52     i = 0;
53     do {
54         cout << "i = " << i << endl;
55         i++;
56     } while (i < 10);
57     cout << endl;
58
59     i = 0;
60     cout << "Do-while loop without increment:" << endl;
61     do {
62         cout << "i = " << i << endl;
63         i++;
64     } while (i < 10);
65
66     return 0;
67 }
```

OUTPUT:

```
(base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1"
&& "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/"Q3
Standard for loop:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

For loop without initialization:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

For loop without increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

For loop without initialization or increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
```

```
Standard while loop:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

While loop without increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

Standard do-while loop:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9

Do-while loop without increment:
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
(base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 %
```

4. Create a structure in C++ containing the details of Students as details below and a main function to execute the structure.

Data Members(properties):

*Name
Roll No
Degree
Hostel
CurrentCGPA*

Member Function(behavior):

*addDetails();
updateDetails();
updateCGPA();
updateHostel();
displaydetails();*

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct Student {
6     string name;
7     int rollNo;
8     string degree;
9     string hostel;
10    float currentCGPA;
11
12    void addDetails() {
13        cout << "Enter student's name: ";
14        cin>>name;
15
16        cout << "Enter roll number: ";
17        cin >> rollNo;
18
19        cout << "Enter degree: ";
20        cin>>degree;
21
22        cout << "Enter hostel: ";
23        cin>>hostel;
24
25        cout << "Enter current CGPA: ";
26        cin >> currentCGPA;
27    }
28
29    void updateDetails() {
```

```
29     void updateDetails() {
30         cout << "Update student's name: ";
31         cin>>name;
32         cout << "Update roll number: ";
33         cin >> rollNo;
34
35         cout << "Update degree: ";
36         cin>>degree;
37
38         cout << "Update hostel: ";
39         cin>>hostel;
40     }
41
42     void updateCGPA() {
43         cout << "Update CGPA: ";
44         cin >> currentCGPA;
45     }
46
47     void updateHostel() {
48         cout << "Update hostel: ";
49         cin>>hostel;
50     }
51
52     void displayDetails() {
53         cout << "\nStudent Details:" << endl;
54         cout << "Name: " << name << endl;
55         cout << "Roll Number: " << rollNo << endl;
56         cout << "Degree: " << degree << endl;
57         cout << "Hostel: " << hostel << endl;
58         cout << "Current CGPA: " << currentCGPA << endl;
59     }
60 };
61
62 int main() {
63     Student student1;
64
65     student1.addDetails();
66
67     student1.displayDetails();
68
69     cout << "\nUpdating student details...\n";
70     student1.updateDetails();
71
72     student1.displayDetails();
73
74     cout << "\nUpdating CGPA...\n";
75     student1.updateCGPA();
76
77     student1.displayDetails();
78
79     cout << "\nUpdating hostel...\n";
80     student1.updateHostel();
81
82
83     student1.displayDetails();
84
85     return 0;
86 }
```

OUTPUT:

```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Desktop/CPP/ASSIGNMENT1/"Q4
Enter student's name: Yash
Enter roll number: 12
Enter degree: coe
Enter hostel: b
Enter current CGPA: 8

Student Details:
Name: Yash
Roll Number: 12
Degree: coe
Hostel: b
Current CGPA: 8

Updating student details...
Update student's name: Yaksh
Update roll number: 2345
Update degree: ECE
Update hostel: F

Student Details:
Name: Yaksh
Roll Number: 2345
Degree: ECE
Hostel: F
Current CGPA: 8

Updating CGPA...
Update CGPA: 7.8

Student Details:
Name: Yaksh
Roll Number: 2345
Degree: ECE
Hostel: F
Current CGPA: 7.8

Updating hostel...
Update hostel: H

Student Details:
Name: Yaksh
Roll Number: 2345
Degree: ECE
Hostel: H
Current CGPA: 7.8
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 %
```

5. Differentiate between private and public access/scope. Perform the question no. 4 with class instead of structure with having the data members private and some member functions in private scope and some in public scope.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 class Student {
6     private:
7         string name;
8         int rollNo;
9         string degree;
10        string hostel;
11        float currentCGPA;
12
13    public:
14
15    void addDetails() {
16        cout << "Enter student's name: ";
17        cin>>name;
18
19        cout << "Enter roll number: ";
20        cin >> rollNo;
21
22        cout << "Enter degree: ";
23        cin>>degree;
24
25        cout << "Enter hostel: ";
26        cin>>hostel;
27
28        cout << "Enter current CGPA: ";
29        cin >> currentCGPA;
30    }
31
32    void updateDetails() {
33        cout << "Update student's name: ";
34        cin>>name;
35        cout << "Update roll number: ";
36        cin >> rollNo;
37        cout << "Update degree: ";
38        cin>>degree;
39        cout << "Update hostel: ";
40        cin>>hostel;
41    }

```

```
44     void updateCGPA() {
45         cout << "Update CGPA: ";
46         cin >> currentCGPA;
47     }
48 private:
49
50     void updateHostel() {
51         cout << "Update hostel: ";
52         cin>>hostel;
53     }
54 public:
55
56     void displayDetails() {
57         cout << "\nStudent Details:" << endl;
58         cout << "Name: " << name << endl;
59         cout << "Roll Number: " << rollNo << endl;
60         cout << "Degree: " << degree << endl;
61         cout << "Hostel: " << hostel << endl;
62         cout << "Current CGPA: " << currentCGPA << endl;
63     }
64 };
65
66 int main() {
67     Student student1;
68
69     student1.addDetails();
70
71     student1.displayDetails();
72
73     cout << "\nUpdating student details...\n";
74     student1.updateDetails();
75
76     student1.displayDetails();
77
78     cout << "\nUpdating CGPA...\n";
79     student1.updateCGPA();
80
81     student1.displayDetails();
82
83     return 0;
84 }
```

OUTPUT:

```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Des  
Enter student's name: Yaksh  
Enter roll number: 1024030280  
Enter degree: COE  
Enter hostel: B  
Enter current CGPA: 7.8  
  
Student Details:  
Name: Yaksh  
Roll Number: 1024030280  
Degree: COE  
Hostel: B  
Current CGPA: 7.8  
  
Updating student details...  
Update student's name: Yashh  
Update roll number: 12034  
Update degree: ECE  
Update hostel: J  
  
Student Details:  
Name: Yashh  
Roll Number: 12034  
Degree: ECE  
Hostel: J  
Current CGPA: 7.8  
  
Updating CGPA...  
Update CGPA: 8.2  
  
Student Details:  
Name: Yashh  
Roll Number: 12034  
Degree: ECE  
Hostel: J  
Current CGPA: 8.2  
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 %
```

6. Create a code snippet that illustrates the following:

- Calling of private member functions inside public member function
- Access private member functions inside public member function

```

1 #include <iostream>
2 using namespace std;
3 ~ class Hostel{
4     private:
5         string name;
6         float keyword;
7
8 ~     void hostelFunction(){
9             cout<<"hostel keyword is"<<keyword<<endl;
10 }
11 public:
12 ~ Hostel(float roomNum){
13     keyword = roomNum;
14 }
15 ~ void hosteler(){
16     cout<<"private data in public function"<<endl;
17     cout<<"the keyword is "<<keyword<<" (accessed directly)"<<endl;
18     hostelFunction();
19 }
20 };
21 ~ int main(){
22     Hostel obj(45);
23     obj.hosteler();
24     return 0;
25 }

```

OUTPUT:

- (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yaksh
IGNMENT1/"Q6

private data in public function

the keyword is 45 (accessed directly)

hostel keyword is45
- (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % []

7. Define a class named ***Complex*** with properties (real and imaginary) and methods as per following details.

void set () to initialize object values.

void display () to display complex number.

Complex sum (Complex) or ***void sum (Complex)*** to add two complex numbers (objects of Complex class) and ***return complex_number*** (object of Complex class) as result.

Properties (real and imaginary) of the code should have private access modifier and member functions should have public access modifier in C++ class.

```
1 #include <iostream>
2 using namespace std;
3 class Complex {
4 private:
5     float real;
6     float imaginary;
7 public:
8     void set(float r, float i) {
9         real = r;
10        imaginary = i;
11    }
12    void display() {
13        cout << real << " + " << imaginary << "i" << endl;
14    }
15
16    Complex sum(Complex c) {
17        Complex temp;
18        temp.real = real + c.real;
19        temp.imaginary = imaginary + c.imaginary;
20        return temp;
21    }
22};
23
24 int main () {
25     Complex c1, c2, c3 ;
26
27     c1.set( 3.5, 2.5);
28     c2.set(1.5, 4.0);
29
30     cout << "First: ";
31     c1.display();
32
33     cout << "Second: ";
34     c2.display();
35
36
37     cout << " Sum:   ";
38     c3.display();
39
40
41     return 0;
42 }
```

OUTPUT:

```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Assignment1/"Q7
First: 3.5 + 2.5i
Second: 1.5 + 4i
Sum: 5 + 6.5i
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 %
```

8. Implement *namespace* in a program to illustrate the use of same name variables and functions in different sections/libraries of the code.

```
1 #include <iostream>
2 using namespace std;
3 namespace Yaksh {
4     float Age=19;
5
6     void getAge(){
7
8         cout << Age;
9
10    }
11 }
12 int main(){
13     float Age;
14     Yaksh::getAge();
15     return 0;
16 }
```

```
● (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 % cd "/Users/yakshgupta/Assignment1/"Q8
19%
○ (base) yakshgupta@Yakshs-MacBook-Air ASSIGNMENT1 %
```