

Lab-2

Topic: Simple C++ programs using Classes and Objects

- i. WAP to display the message "hello" followed by your name on screen.

```
1 //1. Print Hello Followed by your Name
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     char name_167[50];
10    cout << "Enter your Name: ";
11    cin.getline(name_167, 50);
12    cout << "Hello " << name_167 << endl;
13    return 0;
14 }
15
```

```
--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter your Name: Ankit
Hello Ankit
PS T:\KIIT\OOP\OOP LAB>
```

- ii. Create a class which stores name, roll number and total marks for a student. Input the data for a student and display it.

```
1 //2. Store Student data and Display
2
3 #include <iostream>
4
5 using namespace std;
6
7 class student
8 {
9     char name_167[50];
10    int roll_167;
11    int total_167;
12
13 public:
14     void setdata();
15     void display();
16 };
17
18 void student::setdata()
19 {
20     cout << "Enter your Name: ";
21     cin.getline(name_167, 50);
22     cout << "Enter Roll No.: ";
23     cin >> roll_167;
24     cout << "Enter Total Marks: ";
25     cin >> total_167;
26 }
27
28 void student::display()
29 {
30     cout << "Name: " << name_167 << endl;
31     cout << "Roll No.: " << roll_167 << endl;
32     cout << "Total Marks: " << total_167 << endl;
33 }
34
35 int main()
36 {
37     student r;
38     r.setdata();
39     r.display();
40     return 0;
41 }
42
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS T:\KIIT\OOP\OOP LAB> & 'c:\Users\KIIT\In-vn25comg.vfo' '--stdout=Microsoft-MIEngi
--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--
Enter your Name: Ankit
Enter Roll No.: 2006167
Enter Total Marks: 1727
Name: Ankit
Roll No.: 2006167
Total Marks: 1727
PS T:\KIIT\OOP\OOP LAB> █
```

iii. Modify the program ii) to store marks in 5 subjects. Calculate the total marks and percentage of a student and display it.

```
1 //3. Modify 2nd Program to store 5 subjects, total marks, percentage
2
3 #include <iostream>
4
5 using namespace std;
6
7 class student
8 {
9     char name_167[50];
10    int roll_167;
11    int marks_167[5];
12    int total_167;
13    int percentage_167;
14
15 public:
16     void setdata();
17     void display();
18 };
19
20 void student::setdata()
21 {
22     cout << "Enter your Name: ";
23     cin.getline(name_167, 50);
24     cout << "Enter Roll No.: ";
25     cin >> roll_167;
26     total_167 = 0;
27     cout << "Enter Marks in 5 Subjects: " << endl;
28     for (int i = 0; i < 5; i++)
29     {
30         cout << "Mark: ";
31         cin >> marks_167[i];
32         total_167 += marks_167[i];
33     }
34     percentage_167 = total_167 / 5;
35 }
36
37 void student::display()
38 {
39     cout << "Name: " << name_167 << endl;
40     cout << "Roll No.: " << roll_167 << endl;
41     cout << "Total Marks: " << total_167 << " out of 500" << endl;
42     cout << "Percentage: " << percentage_167 << "%" << endl;
43 }
44
45 int main()
46 {
47     student r;
48     r.setdata();
49     r.display();
50     return 0;
51 }
52
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS T:\KIIT\OOP\OOP LAB> & 'c:\Users\KIIT\.vscode\extensions\ms-vscode.cpptool
In-vn25comg.vfo' '--stdout=Microsoft-MIEngine-Out-yeswi2c.clt' '--stderr=Micr
--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter your Name: Ankit
Enter Roll No.: 2006167
Enter Total Marks: 1727
Name: Ankit
Percentage: 92%
PS T:\KIIT\OOP\OOP LAB> □
```

- iv. Create a class complex which stores real and imaginary part of a complex number.
Input 10 complex numbers and display them.

```
1 //4. Complex no. input 10 complex num, and print them
2
3 #include <iostream>
4
5 using namespace std;
6
7 class complex_167
8 {
9     int real_167;
10    int img_167;
11
12 public:
13     void setdata();
14     void display();
15 };
16
17 void complex_167::setdata()
18 {
19     cout << "Enter Real Part: ";
20     cin >> real_167;
21     cout << "Enter Imaginary Part: ";
22     cin >> img_167;
23 }
24
25 void complex_167::display()
26 {
27     cout << real_167 << " + i" << img_167 << endl;
28 }
29
```

```
30 int main()
31 {
32     complex_167 r[3];
33     for (int i = 0; i < 3; i++)
34     {
35         r[i].setdata();
36     }
37     for (int i = 0; i < 3; i++)
38     {
39         r[i].display();
40     }
41     return 0;
42 }
43
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
Percentage: 92%			
PS T:\KIIT\OOP\OOP LAB> & 'c:\Users\KIIT\.vs			
In-jnpvcfcv.ufr' '--stdout=Microsoft-MIEngine			
--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--in			
Enter Real Part:			
PS T:\KIIT\OOP\OOP LAB> & 'c:\Users\KIIT\.vs			
In-lrwp4ym3.xsi' '--stdout=Microsoft-MIEngine			
--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--in			
Enter Real Part: 10			
Enter Imaginary Part: 3			
Enter Real Part: 1			
Enter Imaginary Part: 2			
Enter Real Part: 7			
Enter Imaginary Part: 3			
10 + i3			
1 + i2			
7 + i3			
PS T:\KIIT\OOP\OOP LAB> █			

- v. Create a class `distance` which stores a distance in feet and inches. Input 2 distance values in objects, add them, store the resultant distance in an object and display it.

```
1 //5. feet-inch
2
3 #include <iostream>
4
5 using namespace std;
6
7 class distanceft
8 {
9     int feet_167;
10    int inch_167;
11
12 public:
13     void add(distanceft d1, distanceft d2)
14     {
15         feet_167 = d1.feet_167 + d2.feet_167;
16         inch_167 = d1.inch_167 + d2.inch_167;
17     }
18     distanceft return_add(distanceft d)
19     {
20         distanceft r;
21         r.feet_167 = feet_167 + d.feet_167;
22         r.inch_167 = inch_167 + d.inch_167;
23         return r;
24     }
25     void getinput();
26     void display();
27 };
28
29 void distanceft::getinput()
30 {
31     cout << "Enter feet: ";
32     cin >> feet_167;
33     cout << "Enter inch: ";
34     cin >> inch_167;
35 }
36
```

```

37 void distanceft::display()
38 {
39     if (inch_167 / 12 >= 1)
40     {
41         feet_167 += inch_167 / 12;
42         inch_167 -= 12 * (inch_167 / 12);
43     }
44     cout << feet_167 << "ft " << inch_167 << "inch\n";
45 }
46
47 int main()
48 {
49     distanceft x;
50     distanceft y;
51     x.getinput();
52     y.getinput();
53
54     //Part (a)
55     distanceft z;
56     z.add(x, y);
57     cout << endl
58     | << "Part a:" << endl;
59     z.display();
60
61     //Part (b)
62     distanceft q;
63     q = x.return_add(y);
64     cout << endl
65     | << "Part b:" << endl;
66     q.display();
67     return 0;
68 }

```

--dbgExe=C:\msys64\mingw64\bin\

```

Enter feet: 3
Enter inch: 11
Enter feet: 2
Enter inch: 5

```

```

Part a:
6ft 4inch

```

```

Part b:
6ft 4inch

```

PS T:\KIIT\OOP\OOP LAB> █

- vi. Create a class which stores id, name, age and basic salary of an employee. Input data for n number of employees. Calculate the gross salary of all the employees and display it along with all other details in a tabular form.

[Gross salary= Basic salary + DA + HRA,

DA = 80% of Basic salary

HRA=10% of Basic salary]

```
1 //6. Emp, gross salary
2
3 #include <iostream>
4 using namespace std;
5
6 class emp
7 {
8     char id_167[50];
9     char name_167[50];
10    int age_167;
11    long int pay_167;
12    long int gross_167;
13
14 public:
15     void getdata();
16     void display();
17 };
18
19 void emp::getdata()
20 {
21
22     cin.ignore();
23     cout << "ID: ";
24     gets(id_167);
25     cout << "Name: ";
26     gets(name_167);
27     cout << "Age: ";
28     cin >> age_167;
29     cout << "pay: ";
30     cin >> pay_167;
31 }
32
```

```

33 void emp::display()
34 {
35
36     int hr = (pay_167)*0.8;
37     int dr = (pay_167)*0.1;
38     gross_167 = pay_167 + hr + dr;
39
40     cout << endl
41         << "Display Detail: " << endl
42         << endl;
43
44     cout << "ID: " << id_167 << endl
45         << "Name: " << name_167 << endl;
46     cout << "Age: " << age_167 << endl
47         << "Base Pay: " << pay_167 << endl;
48     cout << "Gross: " << gross_167 << endl
49         << endl;
50 }
51
52 int main()
53 {
54     int n_167;
55     cout << "Enter no of Employee: ";
56     cin >> n_167;
57     emp s[n_167];
58     for (int i = 0; i < n_167; i++)
59     {
60         s[i].getdata();
61     }
62     for (int i = 0; i < n_167; i++)
63     {
64         s[i].display();
65     }
66     return 0;
67 }
68

```

Display Detail:

ID: 2006167@kiit.ac.in
 Name: Ankit
 Age: 18
 Base Pay: 101
 Gross: 191

Display Detail:

ID: 27112002
 Name: Turi
 Age: 18
 Base Pay: 101
 Gross: 191

PS T:\KIIT\OOP\OOP LAB> █

vii. Create a class which stores x and y coordinates of a point. Calculate distance between two given points and display it.

Double calc(point o1, point o2);

a) class point{ Int x, y;

Double calc(point p, point q){

Double dis;

dis=sqrt((p.x-q.x)*(p.x-q.x)+(p.y-q.y)*(p.y-q.y));

Return dis;

```
1 //7. Distance btw oordinates
2
3 #include <iostream>
4 #include <math.h>
5
6 using namespace std;
7
8 class dist
9 {
10     int x_167;
11     int y_167;
12
13 public:
14     double dist_formu(dist g);
15     void get();
16 };
17
18 void dist::get()
19 {
20     cout << "x: ";
21     cin >> x_167;
22     cout << "y: ";
23     cin >> y_167;
24 }
25
26 double dist::dist_formu(dist g)
27 {
28     double gap_167;
29     gap_167 = sqrt((x_167 - g.x_167) * (x_167 - g.x_167) + (y_167 - g.y_167) * (y_167 - g.y_167));
30     cout << "Distance btw the coordinates: ";
31     return gap_167;
32 }
33 }
```

```
34 int main()
35 {
36     dist x, y;
37     x.get();
38     y.get();
39     cout << x.dist_formu(y);
40     return 0;
41 }
42
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS T:\KIIT\OOP\OOP LAB> & 'c:\Users\KIIT\In-sgghcv55.3vl' '--stdout=Microsoft-MIEngi
--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--
x: 2
y: 7
x: 3
y: 10
Distance btw the coordinates: 3.16228
PS T:\KIIT\OOP\OOP LAB> █

viii. Define a class to represent a bank account. Include the following members:

Data Members

b) Name of the depositor

b) Account number

c) Type of account

c) d) Balance **amount** in the account

Member Functions

a) To assign initial value

b) To deposit an amount

c) To withdraw an amount after checking the balance

d) To display name and balance

Write a main program to test the program.

```
1 //8. bank account deposit,withdraw,bal enquiry.
2
3 #include <iostream>
4
5 using namespace std;
6
7 class account
8 {
9     char name_167[50];
10    int acc_no_167;
11    char acc_type_167[20];
12    int balance_167;
13
14 public:
15     void setvalue();
16     void deposit();
17     void withdraw();
18     void display();
19 };
20
21 void account::setvalue()
22 {
23     cout << "Enter Name: ";
24     cin.getline(name_167, 50);
25     cout << "Enter Account Number: ";
26     cin >> acc_no_167;
27     cin.ignore();
28     cout << "Enter Acc. Type: ";
29     cin.getline(acc_type_167, 20);
30 }
31 void account::deposit()
32 {
33     cout << "Enter Balance: ";
34     cin >> balance_167;
35 }
36
```

```

37 void account::withdraw()
38 {
39     int x;
40     cout << "Enter Amount to Widthdraw: ";
41     cin >> x;
42     balance_167 -= x;
43 }
44
45 void account::display()
46 {
47     cout << "Name: " << name_167 << endl;
48     cout << "BAalance: " << balance_167 << endl;
49 }
50
51 int main()
52 {
53     account a;
54     a.setvalue();
55     a.deposit();
56     a.withdraw();
57     a.display();
58     return 0;
59 }
60

```

```

--dbgExe-C:\msys64\mingw64\bin\gdb.exe
Enter Name: Ankit
Enter Account Number: 17112000
Enter Acc. Type: Savings
Enter Balance: 10000
Enter Amount to Widthdraw: 9999
Name: Ankit
BAalance: 1
PS T:\KIIT\OOP\OOP LAB>

```