

CSE-284: Object Oriented Programming

Experiment 4: Inheritance in C++.

Turja Roy
ID: 2108052
Group: G-2

Objectives:

- Familiarization with Inheritance.
- To understand the concept of Single and Multi level inheritance in OOP.
- To solve various problems in order to comprehend the topics.

Example 1

A C++ program to demonstrate the single level inheritance.

Code

```
1 #include <iostream>
2 using namespace std;
3
4 class Shape {
5 protected:
6     int width, height;
7
8 public:
9     void setWidth (int width) {
10         this->width = width;
11     }
12     void setHeight (int height) {
13         this->height = height;
14     }
15 };
16
17 class Rectangle: public Shape {
18 public:
19     int getArea () {
20         return (width * height);
21     }
22 };
23
24 int main ()
25 {
26     Rectangle rect;
27     rect.setWidth(5);
28     rect.setHeight(7);
```

```

29
30     cout << "Total area: " << rect.getArea() << endl;
31
32     return 0;
33 }

```

Output



```

Exp-1.cpp 30:53
2 Total area: 35
1
3 [Process exited 0]

```

Figure 1: Output of Exp-1.cpp

Example 2

A C++ program to demonstrate the Multilevel Inheritance.

Code

```

1  #include <iostream>
2  using namespace std;
3
4  class base {
5  public:
6      int x;
7      void getdata () {
8          cout << "Enter value of x: "; cin >> x;
9      }
10 };
11
12 class derive1 : public base {
13 public:
14     int y;
15     void readdata () {
16         cout << "\nEnter value of y: "; cin >> y;
17     }
18 };
19
20 class derive2 : public derive1 {
21 private:
22     int z;
23
24 public:
25     void indata () {
26         cout << "\nEnter value of z: "; cin >> z;
27     }
28     void product () {
29         cout << "\nProduct: " << x * y * z << endl;
30     }
31 };
32
33 int main ()
34 {
35     derive2 a;
36
37     a.getdata();
38     a.readdata();

```

```

39     a.indata();
40     a.product();
41
42     return 0;
43 }

```

Output

```

Exp-2.cpp 29:50
8 Enter value of x: 5
7
6 Enter value of y: 6
5
4 Enter value of z: 10
3
2 Product: 300
1
9 [Process exited 0]

```

Figure 2: Output of Exp-2.cpp

Practice 1

Write a C++ program to add two numbers. Accept these two numbers from the user in base class and display the sum of these two numbers in derived class.

Code

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 class base {
5 protected:
6     int num1, num2;
7 public:
8     base (int num1, int num2) {
9         this->num1 = num1;
10        this->num2 = num2;
11    }
12 };
13
14 class adder : private base {
15 private:
16     int sum;
17 public:
18     adder (int num1, int num2) : base(num1, num2) {
19         sum = num1 + num2;
20     }
21
22     int getSum () {
23         return sum;
24     }
25 };
26
27 int main ()
28 {
29     adder a(5, 10);
30
31     cout << "Sum: " << a.getSum() << endl;

```

```

32
33     return 0;
34 }

```

Output



```

Prac-1.cpp 21:1
2 Sum: 15
1
3 [Process exited 0]

```

Figure 3: Output of Prac-1.cpp

Practice 2

Write a C++ program to calculate the percentage of a student. Accept the marks of five subjects (Physics, Chemistry, Math, Biology, and English) in base class. A class will be derived from the base class which includes a function to find the total marks obtained and another class derived from this first derived class which calculates and displays the percentage of student.

Hints: Use array for taking the marks of a student.

Code

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  class base {
5  protected:
6      vector<double> marks;
7      void init (vector<double> marks) {
8          this->marks = marks;
9      }
10 };
11
12 class percentage : private base {
13 public:
14     percentage () { }
15
16     void setMarks () {
17         vector<double> marks(5);
18         cout << "Enter the marks for Physics: "; cin >> marks[0];
19         cout << "Enter the marks for Chemistry: "; cin >> marks[1];
20         cout << "Enter the marks for Math: "; cin >> marks[2];
21         cout << "Enter the marks for Biology: "; cin >> marks[3];
22         cout << "Enter the marks for English: "; cin >> marks[4];
23
24         init(marks);
25     }
26
27     double getPercentage () {
28         double total = accumulate(marks.begin(), marks.end(), 0);
29         return total / 5;
30     }
31 };
32
33 int main ()

```

```

34 {
35     percentage p;
36     p.setMarks();
37     cout << "Percentage: " << p.getPercentage() << endl;
38
39     return 0;
40 }

```

Output



```

Prac-2.cpp 28:65
7 Enter the marks for Physics: 97
6 Enter the marks for Chemistry: 87
5 Enter the marks for Math: 99
4 Enter the marks for Biology: 83
3 Enter the marks for English: 92
2 Percentage: 91.6
1
8 [Process exited 0]

```

Figure 4: Output of Prac-2.cpp

Discussion

- Inheritance is a powerful feature in C++ that allows a class to inherit properties and behavior from another class.
- Single level inheritance is a type of inheritance where a class inherits properties and behavior from a single class.
- Multi level inheritance is a type of inheritance where a class inherits properties and behavior from another class which in turn inherits from another class.
- Inheritance is a powerful feature that allows code reusability and helps in reducing redundancy.
- In this experiment, various problems were solved to understand the concept of inheritance in C++.
- In Practice-2 problem, the base class constructor was called from the derived class constructor.