

MIT WORLD PEACE UNIVERSITY

Object Oriented Programming with Java and C++  
Second Year B. Tech, Semester 1

---

---

IMPLEMENTATION OF POLYMORPHISM USING C++  
AND JAVA

---

---

PRACTICAL REPORT

Prepared By

Krishnaraj Thadesar  
Cyber Security and Forensics  
Batch A2, PA 20

October 6, 2022

# Contents

<b>1 Aim and Objectives</b>	<b>1</b>
<b>2 Problem Statements</b>	<b>1</b>
2.1 Problem 1 in C++ . . . . .	1
2.2 Problem 2 in Java . . . . .	1
2.3 Probelm 3 in Java . . . . .	1
<b>3 Theory</b>	<b>2</b>
3.1 Concept of Compile time Polymorphism . . . . .	2
3.2 Concept of Run Time Polymorphism . . . . .	2
3.3 Use of Pure Virtual Functions . . . . .	2
<b>4 Platform</b>	<b>2</b>
<b>5 Input</b>	<b>2</b>
<b>6 Output</b>	<b>2</b>
<b>7 Code</b>	<b>3</b>
7.1 C++ Implementation . . . . .	3
7.1.1 C++ Input and Output . . . . .	7
7.2 Java Implementation of Problem 2 . . . . .	9
7.2.1 Java Output for Problem 2 . . . . .	10
7.3 Java Implementation of Problem 3 using Interfaces . . . . .	10
7.3.1 Java Output . . . . .	11
<b>8 Conclusion</b>	<b>13</b>
<b>9 FAQs</b>	<b>13</b>

## **1 Aim and Objectives**

### **Aim**

Implementation of Polymorphism using C++ and Java.

### **Objectives**

1. To understand the use of pure virtual functions.
2. To understand implantation of compile time and run time polymorphism.
3. To learn implementation of method overriding in java.

## **2 Problem Statements**

### **2.1 Problem 1 in C++**

Write a C++ program with base class Employee and three derived classes namely

- SalariedEmployees
- CommissionEmployees
- HourlyEmployees

Declare calculateSalary() as a pure virtual function in base class and define it in respective derived classes to calculate salary of an employee. The company wants to implement an Object Oriented Application that performs its payroll calculations polymorphically.

### **2.2 Problem 2 in Java**

Define a Class **Shapes** as the Base Class that can find the area of the following :

- Circle
- Square
- Rectangle

Find the area of these shapes using constructor overloading and method overloading.

### **2.3 Problem 3 in Java**

Create a Parent Class Hillstations with the methods location() and famousfor(). Create three subclasses by Hill Station names. These subclasses must extend the superclass and override its methods location() and famousfor(). It should refer to the base class object and the base class method overrides the superclass method, and the base class method is invoked at runtime.

### **3 Theory**

#### **3.1 Concept of Compile time Polymorphism**

#### **3.2 Concept of Run Time Polymorphism**

#### **3.3 Use of Pure Virtual Functions**

### **4 Platform**

**Operating System:** Arch Linux x86-64

**IDEs or Text Editors Used:** Visual Studio Code

**Compilers :** g++ and gcc on linux for C++, and javac, with JDK 18.0.2 for Java

### **5 Input**

#### **For C++**

1. Number of Each Type of Employee
2. Name, Age, Address City, and Salary of Each Employee

#### **For Java**

1. The Side of the Square
2. The Radius of the Circle
3. The Length and Breadth of the Rectangle.

### **6 Output**

#### **For C++**

1. General Information about Each Employee
2. The Weekly, hourly and commisioned Salary for Respective Employees.

#### **For Java**

1. The Area of the Shapes
2. The Location of the Hill Stations
3. The Reason the Hill stations are Famous for.

## 7 Code

### 7.1 C++ Implementation

```
1 // Polymorphism
2 // Virtual Functionss, overriding functions, and overloading functions.
3
4 #include <iostream>
5 using namespace std;
6
7 class Employee
8 {
9 public:
10     // static int ssn;
11     int emp_id = 1000;
12     int age = 0;
13     double basic_sal = 0, da = 0, ta = 0, gross_sal = 0, net_sal = 0;
14     string address_city, position, name;
15
16     virtual void calculate_salary() = 0;
17
18     void display()
19     {
20         // ssn++;
21         // cout << "Employee ssn is:" << ssn << endl;
22         cout << "Employee ID is : " << emp_id << endl;
23         cout << "Employee Name: " << name << endl;
24         cout << "Employee Age: " << age << endl;
25         cout << "Employee Address City: " << address_city << endl;
26     }
27
28     void accept()
29     {
30         cout << "Enter the Employee ID: " << endl;
31         cin >> emp_id;
32         cout << "Enter the Employee Name: " << endl;
33         cin >> name;
34         cout << "Enter the Employee Age: " << endl;
35         cin >> age;
36         cout << "Enter the Employee Address City: " << endl;
37         cin >> address_city;
38     }
39
40     // Destructor
41     ~Employee()
42     {
43         cout << "The Destructor was called" << endl;
44     }
45 };
46 class SalariedEmployee : public Employee
47 {
48 public:
49     int weekly_salary;
50     int net_sal;
51     void accept()
52     {
53         Employee::accept();
54         cout << "Enter the Wage: ";
```

```
55     cin >> weekly_salary;
56 }
57
58 void calculate_salary()
59 {
60     cout << "Calculating Salary of Salaried Employee" << endl;
61     net_sal = weekly_salary * 7;
62 }
63
64 void display()
65 {
66     Employee::display();
67     cout << "Weekly Salary is: " << net_sal << endl;
68 }
69 };
70
71 class HourlyEmployee : public Employee
72 {
73 public:
74     int hours, wage;
75     int net_sal;
76
77     void accept()
78     {
79         Employee::accept();
80         cout << "Enter the basic salary: " << endl;
81         cin >> basic_sal;
82         cout << "Enter the Wage: ";
83         cin >> wage;
84         cout << "Enter the hours worked" << endl;
85         cin >> hours;
86     }
87
88     void calculate_salary()
89     {
90         cout << "Calculating Salary of Salaried Employee" << endl;
91         if (hours < 40)
92         {
93             net_sal = basic_sal + hours * wage;
94         }
95         else
96         {
97             net_sal = 40 * wage + (hours - 40) * wage * 1.5;
98         }
99     }
100
101     void display()
102     {
103         Employee::display();
104         cout << "Hourly Employee Salary is: " << net_sal << endl;
105     }
106 };
107
108 class CommissionEmployee : public Employee
109 {
110 public:
111     float gross_sales, commission_rate = 0.05;
112     float net_sal;
113 }
```

```
114 void accept()
115 {
116     Employee::accept();
117     cout << "Enter the gross sales: ";
118     cin >> gross_sales;
119 }
120
121 void calculate_salary()
122 {
123     cout << "Calculating Salary of Salaried Employee" << endl;
124     net_sal = commission_rate * gross_sales;
125 }
126
127 void display()
128 {
129     Employee::display();
130     cout << "Commission Employee Salary is: " << net_sal << endl;
131 }
132 };
133
134 int main()
135 {
136     cout << "Welcome to Employee Payroll Management System" << endl
137         << endl;
138
139     int choice = 1, number = 1;
140     Employee *ptr;
141     do
142     {
143         cout << "1. Salaried Employee\n2. Commisisioned Employee\n3. Hourly
Employee\n4. Quit\n";
144         cout << "\n\nWhose Details do you wanna enter? " << endl;
145         cin >> choice;
146
147         if (choice == 1)
148         {
149             cout << "How many SalariedEmployees are we talking? ";
150             cin >> number;
151             SalariedEmployee pr[number];
152             for (int i = 0; i < number; i++)
153             {
154                 cout << "Enter the Information about the Salaried Employee" <<
endl;
155                 pr[i].accept();
156             }
157             cout << "\nHere is their Information and their Pay Slips" << endl;
158             cout << endl
159                 << endl;
160
161             cout << "Salaried Employee" << endl;
162
163             for (int i = 0; i < number; i++)
164             {
165                 cout << "Info and Pay Slip of Salaried Employee " << i + 1 << endl
;
166                 ptr = &pr[i];
167                 ptr->calculate_salary();
168                 pr[i].display();
169                 cout << endl;
```

```
170     }
171 }
172 else if (choice == 2)
173 {
174     cout << "How many Commission Employees are we talking? ";
175     cin >> number;
176     CommissionEmployee tl[number];
177     for (int i = 0; i < number; i++)
178     {
179         cout << "Enter the Information about the Commission Employee " <<
180 i + 1 << endl;
181         tl[i].accept();
182     }
183     cout << "Here is their Information and their Pay Slips" << endl;
184     cout << endl
185         << endl;
186     for (int i = 0; i < number; i++)
187     {
188         cout << "Info and Pay Slip of Commission Employee " << i + 1 <<
189 endl;
190         ptr = &tl[i];
191         ptr->calculate_salary();
192         tl[i].display();
193         cout << endl;
194     }
195     cout << endl
196         << endl;
197 }
198 else if (choice == 3)
199 {
200     cout << "How many Hourly Employees are we talking? ";
201     cin >> number;
202     HourlyEmployee ap[number];
203     for (int i = 0; i < number; i++)
204     {
205         cout << "Enter the Information about the Hourly Employees " << i +
206 1 << endl;
207         ap[i].accept();
208     }
209     cout << "Here is their Information and their Pay Slips" << endl;
210     cout << endl
211         << endl;
212     for (int i = 0; i < number; i++)
213     {
214         cout << "Info and Pay Slip of Hourly Employees" << i + 1 << endl;
215         ptr = &ap[i];
216         ptr->calculate_salary();
217         ap[i].display();
218         cout << endl;
219     }
220     cout << endl
221         << endl;
222 }
223 } while (choice != 4);
224 return 0;
```



224 }

### Listing 1: Main.Cpp

#### 7.1.1 C++ Input and Output

```
1 Welcome to Employee Payroll Management System
2
3 1. Salaried Employee
4 2. Commisisioned Employee
5 3. Hourly Employee
6 4. Quit
7
8
9 Whose Details do you wanna enter?
10 1
11 How many SalariedEmployees are we talking? 1
12 Enter the Information about the Salaried Employee
13 Enter the Employee ID:
14 1001
15 Enter the Employee Name:
16 Tony
17 Enter the Employee Age:
18 35
19 Enter the Employee Address City:
20 NewYork
21 Enter the Wage: 50000
22
23 Here is their Information and their Pay Slips
24
25
26 Salaried Employee
27 Info and Pay Slip of Salaried Employee 1
28 Calculating Salary of Salaried Employee
29 Employee ID is : 1001
30 Employee Name: Tony
31 Employee Age: 35
32 Employee Address City: NewYork
33 Weekly Salary is: 350000
34
35 The Destructor was called
36 1. Salaried Employee
37 2. Commisisioned Employee
38 3. Hourly Employee
39 4. Quit
40
41 Whose Details do you wanna enter?
42 2
43 How many Commission Employees are we talking? 1
44 Enter the Information about the Commission Employee 1
45 Enter the Employee ID:
46 1002
47 Enter the Employee Name:
48 Steve
49 Enter the Employee Age:
50 105
51 Enter the Employee Address City:
52 Queens
53 Enter the gross sales: 500
```

## *OOPJC Assignment 3*

---

```
54 Here is their Information and their Pay Slips
55
56
57 Info and Pay Slip of Commission Employee 1
58 Calculating Salary of Salaried Employee
59 Employee ID is : 1002
60 Employee Name: Steve
61 Employee Age: 105
62 Employee Address City: Queens
63 Commission Employee Salary is: 25
64
65
66
67 The Destructor was called
68 1. Salaried Employee
69 2. Commisisioned Employee
70 3. Hourly Employee
71 4. Quit
72
73
74 Whose Details do you wanna enter?
75 3
76 How many Hourly Employees are we talking? 1
77 Enter the Information about the Hourly Employees 1
78 Enter the Employee ID:
79 1003
80 Enter the Employee Name:
81 Bruce
82 Enter the Employee Age:
83 50
84 Enter the Employee Address City:
85 Space
86 Enter the basic salary:
87 600
88 Enter the Wage: 200
89 Enter the hours worked
90 45
91 Here is their Information and their Pay Slips
92
93
94 Info and Pay Slip of Hourly Employees1
95 Calculating Salary of Salaried Employee
96 Employee ID is : 1003
97 Employee Name: Bruce
98 Employee Age: 50
99 Employee Address City: Space
100 Hourly Employee Salary is: 9500
101
102
103
104 The Destructor was called
105 1. Salaried Employee
106 2. Commisisioned Employee
107 3. Hourly Employee
108 4. Quit
109
110
111 Whose Details do you wanna enter?
```

### Listing 2: Output for Problem 1

## 7.2 Java Implementation of Problem 2

```
1 package assignment_3;
2
3 public class Shapes {
4
5     public double Area;
6     public double side;
7     public double length;
8     public double breadth;
9     public int radius;
10
11     Shapes(int radius) {
12         Area = 0.0;
13         this.radius = radius;
14     }
15
16     Shapes(double length, double breadth) {
17         Area = 0.0;
18         this.length = length;
19         this.breadth = breadth;
20     }
21
22     Shapes(double side) {
23         Area = 0.0;
24         this.side = side;
25     }
26
27     double Area(int radius) {
28         Area = 3.14 * radius * radius;
29         return Area;
30     }
31
32     double Area(double length, double breadth) {
33         Area = length * breadth;
34         return Area;
35     }
36
37     double Area(double side) {
38         Area = side * side;
39         return Area;
40     }
41
42 }
```

### Listing 3: Full Time Employee.java

```
1 package assignment_3;
2
3 import java.lang.Math;
4
5 public class Problem_A {
6     public static void main(String[] args) {
7         Shapes circle = new Shapes(7);
8         Shapes square = new Shapes(1.5);
```

```
9      Shapes rectangle = new Shapes(1.4, 3.5);
10      System.out.println("The Radius of the Circle is: " + circle.radius);
11      System.out.println("The Area of the Circle is: " + circle.Area(circle.
    radius));
12      System.out.println("The Side of the Square is: " + square.side);
13      System.out.println("The Area of the Square is: " + square.Area(square.side
    ));
14      System.out.println("The Length of the Rectangle is: " + rectangle.length);
15      System.out.println("The Breadth of the Rectangle is: " + rectangle.breadth
    );
16      System.out.println("The Area of the Rectangle is: "
17          + String.format("%.2f", rectangle.Area(rectangle.length, rectangle
    .breadth)));
18  }
19 }
```

Listing 4: Main.java

### 7.2.1 Java Output for Problem 2

```
1 The Radius of the Circle is: 7
2 The Area of the Circle is: 153.86
3 The Side of the Square is: 1.5
4 The Area of the Square is: 2.25
5 The Length of the Rectangle is: 1.4
6 The Breadth of the Rectangle is: 3.5
7 The Area of the Rectangle is: 4.90
```

Listing 5: Output for Problem 2

### 7.3 Java Implementation of Problem 3 using Interfaces

```
1 package assignment_3;
2
3 abstract public class HillStation {
4     abstract public void location(); // Pure Virtual Function.
5
6     abstract public void famousfor();
7 }
8
9 class Manali extends HillStation {
10     @Override
11     public void location() {
12         System.out.println("Manali in Himachal Pradesh");
13     }
14
15     @Override
16     public void famousfor() {
17         System.out.println(
18             "Manali is a high-altitude Himalayan resort town in India's
19             northern Himachal Pradesh state. It has a reputation as a backpacking center.
20             Set on the Beas River, it's a gateway for skiing in the Solang Valley and
21             trekking in Parvati Valley. It's also a jumping-off point for paragliding,
22             rafting and mountaineering in the Pir Panjal mountains, home to 4,000m-high
23             Rohtang Pass.");
24     }
25 }
26
27 class Shimla extends HillStation {
28     @Override
```

```
24     public void location() {
25         System.out.println("Shimla is in Himachal Pradesh");
26     }
27
28     @Override
29     public void famousfor() {
30         System.out.println(
31             "The town is famous for pleasant walking experiences on hillsides
surrounded by pine and oak forests. This capital city of Himachal Pradesh is
famous for The Mall, ridge, and toy train. With colonial style buildings, the
town has relics of ancient past that lend it a distinct look.");
32     }
33 }
34
35 class Mahabaleshwar extends HillStation {
36     @Override
37     public void location() {
38         System.out.println("Mahabaleshwar is in Maharashtra");
39     }
40
41     @Override
42     public void famousfor() {
43         System.out.println(
44             "Mahabaleshwar is a hill station in India's forested Western Ghats
range, south of Mumbai. It features several elevated viewing points, such as
Arthur's Seat. West of here is centuries-old Pratapgad Fort, perched atop a
mountain spur. East, Lingmala Waterfall tumbles off a sheer cliff. Colorful
boats dot Venna Lake, while 5 rivers meet at Panch Ganga Temple to the north.");
45     }
46 }
```

Listing 6: HillStation

```
1 package assignment_3;
2 public class Problem_B {
3     public static void main(String[] args) {
4         Manali obj = new Manali();
5         obj.location();
6         obj.famousfor();
7
8         Shimla obj1 = new Shimla();
9         obj1.location();
10        obj1.famousfor();
11
12        Mahabaleshwar obj2 = new Mahabaleshwar();
13        obj2.location();
14        obj2.famousfor();
15    }
16 }
```

Listing 7: Main.java

### 7.3.1 Java Output

```
1 Manali in Himachal Pradesh
2 Manali is a high-altitude Himalayan resort town in India's northern Himachal
Pradesh state. It has a reputation as a backpacking center. Set on the Beas
River, it's a gateway for skiing in the Solang Valley and trekking in Parvati
```

```
Valley. It's also a jumping-off point for paragliding, rafting and
mountaineering in the Pir Panjal mountains, home to 4,000m-high Rohtang Pass.
3 Shimla is in Himachal Pradesh
4 The town is famous for pleasant walking experiences on hillsides surrounded by
  pine and oak forests. This capital city of Himachal Pradesh is famous for The
  Mall, ridge, and toy train. With colonial style buildings, the town has relics
  of ancient past that lend it a distinct look.
5 Mahabaleshwar is in Maharashtra
6 Mahabaleshwar is a hill station in India's forested Western Ghats range, south of
  Mumbai. It features several elevated viewing points, such as Arthur's Seat.
  West of here is centuries-old Pratapgad Fort, perched atop a mountain spur.
  East, Lingmala Waterfall tumbles off a sheer cliff. Colorful boats dot Venna
  Lake, while 5 rivers meet at Panch Ganga Temple to the north.
```

Listing 8: Output for Problem 3

## **8 Conclusion**

Thus, learned to use polymorphism and implemented solution of the given problem statement using C++ and Java.

## **9 FAQs**

1. **Discuss the use of Virtual Functions.**
2. **What is the difference between early binding and late binding.**
3. **Explain the use of abstract keyword in java with examples.**
4. **State Features of abstract base classes.**