**Inheritance**

1. Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call

1 - method of parent class by object of parent class  
2 - method of child class by object of child class  
3 - method of parent class by object of child class

1 class one {

2 void print()

3 {

4 System.out.println("This is parent class");

5 }

6 }

7

8 class two extends one {

9 void print\_two()

10 {

11 System.out.println("This is child class");

12 }

13 }

14

15 class Main {

16 public static void main(String[] args) {

17 one obOne = new one();

18 obOne.print();

19

20 two obj = new two();

21 obj.print\_two();

22 obj.print();

23 }

24 }

1. In the above example, declare the method of the parent class as private and then repeat the first two operations (You will get error in the third).

Illegal modifier for the class one; only public, abstract & final are permittedJava(16777518)

1. Create a class named 'Member' having the following members:  
   Data members  
   1 - Name  
   2 - Age  
   3 - Phone number  
   4 - Address  
   5 – Salary

It also has a method named 'printSalary' which prints the salary of the members.  
Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

1 class Memeber {

2 String name, address;

3 int age, phone\_number, salary;

4

5 void printSalary() {

6 System.out.println("Salary: " + salary);

7 }

8 }

9

10 class Employee extends Memeber {

11 String specialization;

12 }

13

14 class Manager extends Memeber {

15 String department;

16 }

17

18 public class two {

19 public static void main(String[] args) {

20 Employee employee = new Employee();

21 Manager manager = new Manager();

22

23 employee.name = "Employee";

24 employee.address = "Employee Address";

25 employee.age = 20;

26 employee.phone\_number = 1234567890;

27 employee.salary = 10000;

28 employee.specialization = "Employee Specialization";

29

30 manager.name = "Manager";

31 manager.address = "Manager Address";

32 manager.age = 30;

33 manager.phone\_number = 1234567890;

34 manager.salary = 20000;

35 manager.department = "Manager Department";

36

37 System.out.println("Employee Details:");

38 System.out.println("Name: " + employee.name);

39 System.out.println("Address: " + employee.address);

40 System.out.println("Age: " + employee.age);

41 System.out.println("Phone Number: " + employee.phone\_number);

42 employee.printSalary();

43 System.out.println("Specialization: " + employee.specialization);

44

45 System.out.println("\nManager Details:");

46 System.out.println("Name: " + manager.name);

47 System.out.println("Address: " + manager.address);

48 System.out.println("Age: " + manager.age);

49 System.out.println("Phone Number: " + manager.phone\_number);

50 manager.printSalary();

51 System.out.println("Department: " + manager.department);

52 }

53 }

1. Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.

1 class Rectangle {

2 Integer length;

3 Integer width;

4

5 public Rectangle(Integer length, Integer width) {

6 this.length = length;

7 this.width = width;

8 }

9

10 public Integer getArea() {

11 return length \* width;

12 }

13

14 public Integer getPerimeter() {

15 return 2 \* (length + width);

16 }

17 }

18

19 class Square extends Rectangle {

20 public Square(Integer side) {

21 super(side, side);

22 }

23 }

24

25 public class Four {

26 public static void main(String[] args) {

27 Rectangle rectangle = new Rectangle(10, 20);

28 System.out.println("Rectangle Area: " + rectangle.getArea());

29 System.out.println("Rectangle Perimeter: " + rectangle.getPerimeter());

30

31 Square square = new Square(10);

32 System.out.println("Square Area: " + square.getArea());

33 System.out.println("Square Perimeter: " + square.getPerimeter());

34 }

35 }

1. Now repeat the above example to print the area of 15 squares.  
   Hint-Use array of objects

1 class Rectangle {

2 Integer length;

3 Integer width;

4

5 public Rectangle(Integer length, Integer width) {

6 this.length = length;

7 this.width = width;

8 }

9

10 public Integer getArea() {

11 return length \* width;

12 }

13

14 public Integer getPerimeter() {

15 return 2 \* (length + width);

16 }

17 }

18

19 class Square extends Rectangle {

20 public Square(Integer side) {

21 super(side, side);

22 }

23 }

24

25 public class Five {

26 public static void main(String[] args) {

27 Square square = new Square(5);

28 System.out.println("Square Area: " + square.getArea());

29 System.out.println("Square Perimeter: " + square.getPerimeter());

30 }

31 }

1. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

1 class Shape {

2 public void draw\_shape() {

3 System.out.println("This is Shape");

4 }

5 }

6

7 class Rectangle extends Shape {

8 public void draw\_rectangle() {

9 System.out.println("This is Rectangluar shape");

10 }

11 }

12

13 class Circle extends Shape {

14 public void draw\_circle() {

15 System.out.println("This is Circular shape");

16 }

17 }

18

19 class Square extends Rectangle {

20 public void draw\_square() {

21 System.out.println("Square is a rectangle");

22 }

23 }

24

25 public class Six {

26 public static void main(String[] args) {

27 Square square = new Square();

28 square.draw\_shape();

29 square.draw\_rectangle();

30 }

31 }

1. Write an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the superclass of the hierarchy. Create and use a Point class to represent the points in each shape. Make the hierarchy as deep (i.e., as many levels) as possible. Specify the instance variables and methods for each class. The private instance variables of Quadrilateral should be the *x-y* coordinate pairs for the four endpoints of the Quadrilateral. Write a program that instantiates objects of your classes and outputs each object’s area (except Quadrilateral).

1 class Point {

2 private int x;

3 private int y;

4

5 public Point(int x, int y) {

6 this.x = x;

7 this.y = y;

8 }

9

10 public int getX() {

11 return x;

12 }

13

14 public int getY() {

15 return y;

16 }

17 }

18

19 class Quadrilateral {

20 private Point point1;

21 private Point point2;

22 private Point point3;

23 private Point point4;

24

25 public Quadrilateral(Point point1, Point point2, Point point3, Point point4) {

26 this.point1 = point1;

27 this.point2 = point2;

28 this.point3 = point3;

29 this.point4 = point4;

30 }

31

32 public Point getPoint1() {

33 return point1;

34 }

35

36 public Point getPoint2() {

37 return point2;

38 }

39

40 public Point getPoint3() {

41 return point3;

42 }

43

44 public Point getPoint4() {

45 return point4;

46 }

47

48 public double A() {

49 return (double) Math.sqrt(Math.pow(getPoint2().getX() - getPoint1().getX(), 2)

50 + Math.pow(getPoint2().getY() - getPoint1().getY(), 2));

51 }

52

53 public double B() {

54 return (double) Math.sqrt(Math.pow(getPoint3().getX() - getPoint2().getX(), 2)

55 + Math.pow(getPoint3().getY() - getPoint2().getY(), 2));

56 }

57

58 public double C() {

59 return (double) Math.sqrt(Math.pow(getPoint4().getX() - getPoint3().getX(), 2)

60 + Math.pow(getPoint4().getY() - getPoint3().getY(), 2));

61 }

62

63 public double D() {

64 return (double) Math.sqrt(Math.pow(getPoint1().getX() - getPoint4().getX(), 2)

65 + Math.pow(getPoint1().getY() - getPoint4().getY(), 2));

66 }

67 }

68

69 class Trapezoid extends Quadrilateral {

70 public Trapezoid(Point point1, Point point2, Point point3, Point point4) {

71 super(point1, point2, point3, point4);

72 }

73

74 public double area() {

75 return 0.5 \* (A() + C()) \* B();

76 }

77 }

78

79 class Parallelogram extends Quadrilateral {

80 public Parallelogram(Point point1, Point point2, Point point3, Point point4) {

81 super(point1, point2, point3, point4);

82 }

83

84 public double area() {

85 return A() \* B();

86 }

87 }

88

89 class Rectangle extends Parallelogram {

90 public Rectangle(Point point1, Point point2, Point point3, Point point4) {

91 super(point1, point2, point3, point4);

92 }

93

94 public double area() {

95 return A() \* B();

96 }

97 }

98

99 class Square extends Rectangle {

100 public Square(Point point1, Point point2, Point point3, Point point4) {

101 super(point1, point2, point3, point4);

102 }

103

104 public double area() {

105 return A() \* B();

106 }

107 }

108

109 public class Seven {

110 public static void main(String[] args) {

111 Point point1 = new Point(0, 0);

112 Point point2 = new Point(0, 5);

113 Point point3 = new Point(5, 5);

114 Point point4 = new Point(5, 0);

115

116 Trapezoid trapezoid = new Trapezoid(point1, point2, point3, point4);

117 Parallelogram parallelogram = new Parallelogram(point1, point2, point3, point4);

118 Rectangle rectangle = new Rectangle(point1, point2, point3, point4);

119 Square square = new Square(point1, point2, point3, point4);

120

121 System.out.println("Area of Trapezoid: " + trapezoid.area());

122 System.out.println("Area of Parallelogram: " + parallelogram.area());

123 System.out.println("Area of Rectangle: " + rectangle.area());

124 System.out.println("Area of Square: " + square.area());

125 }

126 }