3. **public** **class** VisibilityExample {

**public** **int** publicvariable=10;

**private** **int** privatevariable=20;

**public** **void** publicmethod() {

System.***out***.println("This a public method");

}

**private** **void** privatemethod() {

System.***out***.println("This a private method");

}

**public** **int** getprivatevariable() {

**return** privatevariable;

}

**public** **void** setprivatevariable(**int** value) {

**this**.privatevariable=value;

}

**public** **void** callprivatemethod() {

privatemethod();

}

}

**class** Main{

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

VisibilityExample obj=**new** VisibilityExample();

System.***out***.println(obj.publicvariable);

System.***out***.println(obj.getprivatevariable());

obj.setprivatevariable(50);

System.***out***.println(obj.getprivatevariable());

obj.callprivatemethod();

obj.publicmethod();

}

}

4. **import** java.util.Scanner;

**public** **class** weekday {

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

Scanner input = **new** Scanner(System.***in***);

System.***out***.print("Enter number (1-7): ");

**int** num = input.nextInt();

**if** (num == 1) {

System.***out***.println("Monday");

} **else** **if** (num == 2) {

System.***out***.println("Tuesday");

} **else** **if** (num == 3) {

System.***out***.println("Wednesday");

} **else** **if** (num == 4) {

System.***out***.println("Thursday");

} **else** **if** (num == 5) {

System.***out***.println("Friday");

} **else** **if** (num == 6) {

System.***out***.println("Saturday");

} **else** **if** (num == 7) {

System.***out***.println("Sunday");

} **else** {

System.***out***.println("Invalid input! Enter a number between 1 and 7.");

}

input.close();

}

}

5. **import** java.util.Scanner;

**public** **class** mul\_table {

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

Scanner input=**new** Scanner(System.***in***);

System.***out***.println("enter n:");

**int** n=input.nextInt();

**for**(**int** i=0;i<=n;i++) {

System.***out***.println(n\*i);

}

}

}

6. **import** java.util.Scanner;

**public** **class** floating\_point {

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

Scanner input=**new** Scanner(System.***in***);

System.***out***.print("Enter f1:");

**double** f1=input.nextDouble();

System.***out***.print("Enter f2:");

**double** f2=input.nextDouble();

**int** int1=(**int**)(f1\*1000);

**int** int2=(**int**)(f2\*1000);

**if** (int1!=int2) {

System.***out***.println("They are different");

}

**else** {

System.***out***.println("They are same");

}

}}

7. **import** java.util.Scanner;

**public** **class** inc\_dec {

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

Scanner input=**new** Scanner(System.***in***);

System.***out***.println("enter num1:");

**int** num1=input.nextInt();

System.***out***.println("enter num2:");

**int** num2=input.nextInt();

System.***out***.println("enter num3:");

**int** num3=input.nextInt();

**if**(num1>num2 && num2>num3) {

System.***out***.println("Increasing Order");

}

**else** **if**(num1<num2 && num2<num3) {

System.***out***.println("Decreasing Order");

}

**else** {

System.***out***.println("Neither increasing or decreasing order");

}

}

}

8. **import** java.util.Scanner;

**public** **class** DigitCounter {

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

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("Input an integer number less than ten billion: ");

**long** number = scanner.nextLong();

**if** (number <= 0 || number >= 10000000000L) {

System.***out***.println("Number is not within the valid range (1 to 9,999,999,999).");

} **else** {

**int** digits = 0;

**if** (number < 10) {

digits = 1;

} **else** **if** (number < 100) {

digits = 2;

} **else** **if** (number < 1000) {

digits = 3;

} **else** **if** (number < 10000) {

digits = 4;

} **else** **if** (number < 100000) {

digits = 5;

} **else** **if** (number < 1000000) {

digits = 6;

} **else** **if** (number < 10000000) {

digits = 7;

} **else** **if** (number < 100000000) {

digits = 8;

} **else** **if** (number < 1000000000) {

digits = 9;

} **else** {

digits = 10;

}

System.***out***.println("Number of digits in the number: " + digits);

}

scanner.close();

}}

9. **public** **class** PascalsTriangle {

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

**int** rows = 5; // You can change the number of rows here

**for** (**int** i = 0; i < rows; i++) {

// Print leading spaces

**for** (**int** space = 0; space < rows - i; space++) {

System.***out***.print(" ");

}

**int** number = 1;

**for** (**int** j = 0; j <= i; j++) {

System.***out***.print(number + " ");

number = number \* (i - j) / (j + 1); // nCr logic

}

System.***out***.println();

}

}

}

10. **public** **class** char\_rhombhus {

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

**int** rows = 7; // Change this to control the height (center line will have 13 characters)

// Upper half including middle row

**for** (**int** i = 0; i < rows; i++) {

// Leading spaces

**for** (**int** space = 0; space < rows - i - 1; space++) {

System.***out***.print(" ");

}

// Ascending characters: A to current

**for** (**char** ch = 'A'; ch < 'A' + i + 1; ch++) {

System.***out***.print(ch);

}

// Descending characters: current - 1 to A

**for** (**char** ch = (**char**) ('A' + i - 1); ch >= 'A'; ch--) {

System.***out***.print(ch);

}

System.***out***.println();

}

// Lower half

**for** (**int** i = rows - 2; i >= 0; i--) {

// Leading spaces

**for** (**int** space = 0; space < rows - i - 1; space++) {

System.***out***.print(" ");

}

// Ascending characters: A to current

**for** (**char** ch = 'A'; ch < 'A' + i + 1; ch++) {

System.***out***.print(ch);

}

// Descending characters

**for** (**char** ch = (**char**) ('A' + i - 1); ch >= 'A'; ch--) {

System.***out***.print(ch);

}

System.***out***.println();

}

}

}

11. nadi thappu

12. **public** **class** Employee {

**public** **void** work() {

System.***out***.println("Employee is working");

}

**public** **void** getSalary() {

System.***out***.println("You can get the salary by the monthend");

}

}

**class** HRManager **extends** Employee{

**public** **void** work() {

System.***out***.println("HRManager is not working");

}

**public** **void** addEmployee() {

System.***out***.println("employee is added");

}

}

**class** Overriding{

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

HRManager obj=**new** HRManager();

obj.work();

obj.getSalary();

obj.addEmployee();

}

}

13. **class** Calculator {

**int** a = 20, b = 10;

**public** **void** add() {

System.***out***.println("Addition: " + (a + b));

}

**public** **void** subtract() {

System.***out***.println("Subtraction: " + (a - b));

}

}

//First derived class: Extends Calculator

**class** AdvancedCalculator **extends** Calculator {

**public** **void** multiply() {

System.***out***.println("Multiplication: " + (a \* b));

}

}

//Second derived class: Extends AdvancedCalculator (multilevel)

**class** SuperCalculator **extends** AdvancedCalculator {

**public** **void** divide() {

**if** (b != 0) {

System.***out***.println("Division: " + (a / b));

} **else** {

System.***out***.println("Cannot divide by zero.");

}

}

}

//Main class to test the calculator

**class** MultilevelCalculator {

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

SuperCalculator calc = **new** SuperCalculator();

System.***out***.println("Performing Calculator Operations:");

calc.add();

calc.subtract();

calc.multiply();

calc.divide();

}

}

14. **public** **class** Regular\_employee {

String name;

**int** employeeId;

**public** Regular\_employee(String name, **int** employeeId) {

**this**.name = name;

**this**.employeeId = employeeId;

}

**public** **void** salary() {

**int** salary = 200000;

System.***out***.println("Salary of a regular employee is: " + salary);

}

}

**class** Manager **extends** Regular\_employee {

String managerName;

**int** managerId;

**int** bonus;

**public** Manager(String name, **int** employeeId, **int** bonus) {

**super**(name, employeeId);

**this**.managerName = name;

**this**.managerId = employeeId;

**this**.bonus = bonus;

}

**public** **void** salary() {

**int** salary = 300000;

**int** totalSalary = salary + bonus;

System.***out***.println("Salary of a Manager is: " + totalSalary);

}

**public** **void** display() {

System.***out***.println("Name: " + managerName);

System.***out***.println("Employee ID: " + managerId);

// Accessing values from Regular\_employee

System.***out***.println("Regular Employee Name: " + **super**.name);

System.***out***.println("Regular Employee ID: " + **super**.employeeId);

**super**.salary(); // Regular salary

salary(); // Manager's salary

}

}

**class** SingleInheritance {

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

Manager man = **new** Manager("Hasini", 24013, 10000);

man.display();

}

}

15. **public** **class** Shapes {

**public** **void** area() {

System.***out***.println("Calculating area...");

}

}

**class** Triangle **extends** Shapes {

**public** **void** area() {

**int** base = 5;

**int** height = 10;

**double** area = 0.5 \* base \* height;

System.***out***.println("Area of Triangle: " + area);

}

}

**class** Circle **extends** Shapes {

**public** **void** area() {

**int** radius = 7;

**double** pi = 3.14159;

**double** area = pi \* radius \* radius;

System.***out***.println("Area of Circle: " + area);

}

}

**class** Rectangle **extends** Shapes {

**public** **void** area() {

**int** length = 8;

**int** breadth = 6;

**int** area = length \* breadth;

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

}

}

**class** Overridden {

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

Shapes shape;

shape = **new** Triangle();

shape.area();

shape = **new** Circle();

shape.area();

shape = **new** Rectangle();

shape.area();

}

}

16. **public** **class** Animal {

**public** **void** eat() {

System.***out***.println("Animal is eating");

}

}

**class** Herbivores **extends** Animal{

**public** **void** eat() {

System.***out***.println("Herbivores eats plants");

}

}

**class** Carnivores **extends** Animal{

**public** **void** eat() {

System.***out***.println("Carnivores eats animals");

}

}

**class** Omnivores **extends** Animal{

**public** **void** eat() {

System.***out***.println("Omnivores eats both plants and animals");

}

}

**class** q16{

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

Animal goat;

goat=**new** Herbivores();

goat.eat();

Animal Lion;

Lion=**new** Carnivores();

Lion.eat();

Animal Human;

Human=**new** Omnivores();

Human.eat();

}

}

17. **abstract** **class** Animal\_Abst {

**public** **abstract** **void** sound();

}

**class** Lion **extends** Animal\_Abst {

**public** **void** sound() {

System.***out***.println("Lion is groaring");

}

}

**class** Tiger **extends** Animal\_Abst {

**public** **void** sound() {

System.***out***.println("Tiger is roaring");

}

}

**class** Abstractclass {

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

Animal\_Abst obj1;

obj1 = **new** Lion();

obj1.sound();

Animal\_Abst obj2;

obj2 = **new** Tiger();

obj2.sound();

}

}

18. **public** **abstract** **class** Shape3D {

**int** r=8;

**public** **abstract** **void** calculateVolume();

**public** **abstract** **void** calculateSurfaceArea();

}

**class** Sphere **extends** Shape3D{

**public** **void** calculateVolume() {

**double** pi=3.14;

**double** vol=1.33\*pi\*r\*r\*r;

System.***out***.println("volume of sphere is "+vol);

}

**public** **void** calculateSurfaceArea() {

**double** pi=3.14;

**double** A=4\*pi\*r\*r;

System.***out***.println("surface area of sphere is "+A);

}

}

**class** Cube **extends** Shape3D{

**public** **void** calculateVolume() {

**int** a=5;

**int** vol=a^3;

System.***out***.println("volume of cube is "+vol);

}

**public** **void** calculateSurfaceArea() {

**int** a=6;

**int** S\_A=6\*(a^2);

System.***out***.println("Surface area of the cube is: "+S\_A);

}

}

**class** q18{

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

Shape3D obj1;

obj1=**new** Sphere();

obj1.calculateVolume();

obj1.calculateSurfaceArea();

Shape3D obj2;

obj2=**new** Cube();

obj2.calculateVolume();

obj2.calculateSurfaceArea();

}

}

19.OUTPUT: My method

19.

|  |  |
| --- | --- |
| Does it compile? | No |
| Reason | Cannot assign value to final variable j |
| 20. **interface** Shape {  **void** getPerimeter();  }  **class** Rectangle **implements** Shape {  **public** **void** getPerimeter() {  **int** length = 5;  **int** breadth = 7;  **int** perimeter = 2 \* (length + breadth);  System.***out***.println("Perimeter of Rectangle: " + perimeter);  }  }  **class** Circle **implements** Shape {  **public** **void** getPerimeter() {  **int** radius = 4;  **double** pi = 3.14;  **double** perimeter = 2 \* pi \* radius;  System.***out***.println("Perimeter of Circle: " + perimeter);  }  }  **class** Triangle **implements** Shape {  **public** **void** getPerimeter() {  **int** a = 3, b = 4, c = 5;  **int** perimeter = a + b + c;  System.***out***.println("Perimeter of Triangle: " + perimeter);  }  }  **public** **class** Q\_20 {  **public** **static** **void** main(String[] args) {  Shape rect = **new** Rectangle();  rect.getPerimeter();  Shape circle = **new** Circle();  circle.getPerimeter();  Shape triangle = **new** Triangle();  triangle.getPerimeter();  }  }  21. **class** Employee21 {  **protected** String name;  **protected** String address;  **protected** **double** salary;  **protected** String jobTitle;  **public** Employee21(String name, String address, **double** salary, String jobTitle) {  **this**.name = name;  **this**.address = address;  **this**.salary = salary;  **this**.jobTitle = jobTitle;  }  **public** **double** calculateBonus() {  **return** salary \* 0.05; // default 5% bonus  }  **public** String generatePerformanceReport() {  **return** "Performance report for " + name + " (" + jobTitle + "): Satisfactory.";  }  **public** **void** manageProject(String projectName) {  System.***out***.println(name + " is managing the project: " + projectName);  }  **public** **void** displayInfo() {  System.***out***.println("Name: " + name);  System.***out***.println("Address: " + address);  System.***out***.println("Salary: $" + salary);  System.***out***.println("Job Title: " + jobTitle);  }  }  //Manager subclass  **class** Manager **extends** Employee21 {  **public** Manager(String name, String address, **double** salary) {  **super**(name, address, salary, "Manager");  }  @Override  **public** **double** calculateBonus() {  **return** salary \* 0.20; // Managers get 20% bonus  }  @Override  **public** String generatePerformanceReport() {  **return** "Manager " + name + " has exceeded performance expectations.";  }  @Override  **public** **void** manageProject(String projectName) {  System.***out***.println("Manager " + name + " is leading the project: " + projectName);  }  }  //Developer subclass  **class** Developer **extends** Employee21 {  **public** Developer(String name, String address, **double** salary) {  **super**(name, address, salary, "Developer");  }  @Override  **public** **double** calculateBonus() {  **return** salary \* 0.10; // Developers get 10% bonus  }  @Override  **public** String generatePerformanceReport() {  **return** "Developer " + name + " has met all deadlines and delivered clean code.";  }  @Override  **public** **void** manageProject(String projectName) {  System.***out***.println("Developer " + name + " is working on the project: " + projectName);  }  }  //Programmer subclass  **class** Programmer **extends** Employee21 {  **public** Programmer(String name, String address, **double** salary) {  **super**(name, address, salary, "Programmer");  }  @Override  **public** **double** calculateBonus() {  **return** salary \* 0.08; // Programmers get 8% bonus  }  @Override  **public** String generatePerformanceReport() {  **return** "Programmer " + name + " is consistent and productive.";  }  @Override  **public** **void** manageProject(String projectName) {  System.***out***.println("Programmer " + name + " is developing modules for: " + projectName);  }  }  **class** CompanyHierarchy {  **public** **static** **void** main(String[] args) {  Manager mgr = **new** Manager("Alice Smith", "123 Business Rd", 95000);  Developer dev = **new** Developer("Bob Johnson", "456 Tech Ave", 85000);  Programmer prog = **new** Programmer("Charlie Brown", "789 Code Blvd", 75000);  Employee21[] employees = {mgr, dev, prog};  **for** (Employee21 emp : employees) {  emp.displayInfo();  System.***out***.println("Bonus: $" + emp.calculateBonus());  System.***out***.println(emp.generatePerformanceReport());  emp.manageProject("Alpha Project");  System.***out***.println("---------------");  }  }  }  22. **public** **class** Student{  **private** **int** student\_id;  **private** String student\_name;  **private** **char** grade='A';  **public** **void** addGrade() {  **int** grade\_ord=((**int**)grade)+1;  **char** grade\_char=(**char**)(grade\_ord);  System.***out***.println("your grade is promoted to "+grade\_char);  }  **public** **int** getstudent\_id(){  **return** student\_id;  }  **public** String getstudent\_name() {  **return** student\_name;  }  **public** **char** getgrade() {  **return** grade;  }  **public** **void** setstudent\_id(**int** value) {  **this**.student\_id=value;  }  **public** **void** setstudent\_name(String value) {  **this**.student\_name=value;  }  }  **class** q\_22{  **public** **static** **void** main(String[] args) {  Student s=**new** Student();  s.setstudent\_id(24013);  s.setstudent\_name("Hasini");  System.***out***.println(s.getstudent\_id());  System.***out***.println(s.getstudent\_name());  System.***out***.println(s.getgrade());  s.addGrade();  }  } |  |
|  |  |