Name – Harshit

MIS – 112316018

Java Lab

Assignment 7

1. Create a class “Vehicle” with a method “start()” that prints “Vehicle started”. Create a subclass “Car” that extends “Vehicle” and overrides the “start()” method to print “Car started”. Create an object of the “Vehicle” class and call the “start()” method. Create an object of the “Car” class and call the “start()” method.

Code:

class Vehicle {

void start() {

System.out.println("Vehicle started");

}

}

class Car extends Vehicle {

@Override

void start() {

System.out.println("Car started");

}

}

public class main7 {

public static void main(String[] args) {

Vehicle v = new Vehicle();

v.start();

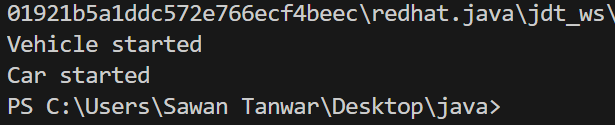
Car c = new Car();

c.start();

}

}

Output:

7

2. Create a class “Person” with fields “name” and “age” and a method “display()” that prints the name and age of the person. Create a subclass “Employee” that extends “Person” and adds a field “salary” and a method “display()” that prints the name, age, and salary of the employee. Create an object of the “Person” class and call the “display()” method. Create an object of the `Employee` class and call the “display()” method.

Code:

class Person {

String name;

int age;

Person(String name, int age) {

this.name = name;

this.age = age;

}

void display() {

System.out.println("Name: " + name + ", Age: " + age);

}

}

class Employee extends Person {

double salary;

Employee(String name, int age, double salary) {

super(name, age);

this.salary = salary;

}

@Override

void display() {

System.out.println("Name: " + name + ", Age: " + age + ", Salary: " + salary);

}

}

public class main7 {

public static void main(String[] args) {

Person p = new Person("Alice", 30);

p.display();

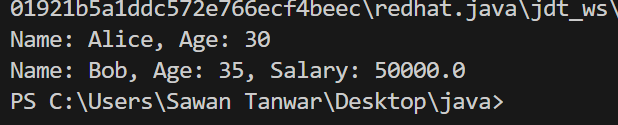
Employee e = new Employee("Bob", 35, 50000);

e.display();

}

}

Output



3. Create a class “Shape” with a method “getArea()” that returns the area of the shape. Create a subclass “Rectangle” that extends “Shape” and adds fields “length” and “width” and overrides the “getArea()” method to return the area of the rectangle. Create an object of the “Shape” class and call the “getArea()” method. Create an object of the “Rectangle” class and call the “getArea()” method.

Code:

class Shape {

double getArea() {

return 0;

}

}

class Rectangle extends Shape {

double length, width;

Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

double getArea() {

return length \* width;

}

}

public class main7 {

public static void main(String[] args) {

Shape s = new Shape();

System.out.println("Shape Area: " + s.getArea());

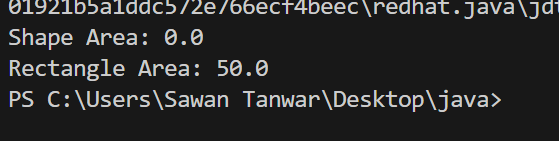
Rectangle r = new Rectangle(5, 10);

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

}

}

Output:



4. Create a class “Animal” with a method “makeSound()” that prints a sound. Create a subclass “Dog” that extends “Animal” and overrides the “makeSound()” method to print “Woof!”. Create an object of the “Animal” class and call the “makeSound()” method. Create an object of the “Dog” class and call the “makeSound()” method.

Code:  
class Animal {

void makeSound() {

System.out.println("Some sound");

}

}

class Dog extends Animal {

@Override

void makeSound() {

System.out.println("Woof!");

}

}

public class main7 {

public static void main(String[] args) {

Animal a = new Animal();

a.makeSound();

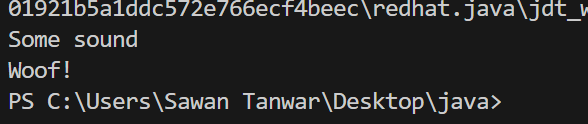
Dog d = new Dog();

d.makeSound();

}

}

Output:



5. Create a class “BankAccount” with fields “accountNumber”, “balance”, and “interestRate” and a method “deposit()” that adds an amount to the balance. Create a subclass “SavingsAccount” that extends “BankAccount” and adds a field “minimumBalance” and a method “withdraw()” that subtracts an amount from the balance. Create an object of the “BankAccount” class and call the “deposit()” method. Create an object of the “SavingsAccount” class and call the “deposit()” and “withdraw()” methods.

Code:  
class BankAccount {

String accountNumber;

double balance;

double interestRate;

BankAccount(String accountNumber, double balance, double interestRate) {

this.accountNumber = accountNumber;

this.balance = balance;

this.interestRate = interestRate;

}

void deposit(double amount) {

balance += amount;

System.out.println("Deposited: " + amount + ", New Balance: " + balance);

}

}

class SavingsAccount extends BankAccount {

double minimumBalance;

SavingsAccount(String accountNumber, double balance, double interestRate, double minimumBalance) {

super(accountNumber, balance, interestRate);

this.minimumBalance = minimumBalance;

}

void withdraw(double amount) {

if (balance - amount >= minimumBalance) {

balance -= amount;

System.out.println("Withdrawn: " + amount + ", New Balance: " + balance);

} else {

System.out.println("Withdrawal denied! Balance cannot go below minimum balance.");

}

}

}

public class main7 {

public static void main(String[] args) {

BankAccount ba = new BankAccount("123456", 1000, 2.5);

ba.deposit(500);

SavingsAccount sa = new SavingsAccount("789012", 2000, 3.0, 500);

sa.deposit(300);

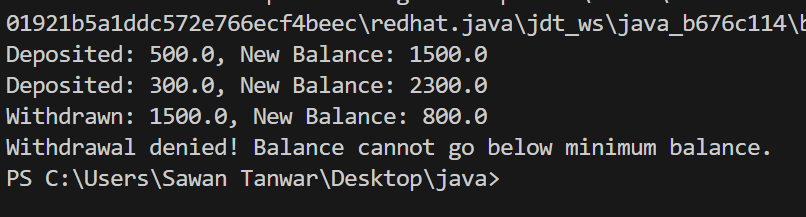
sa.withdraw(1500);

sa.withdraw(2000);

}

}

Output:



6. Create a class “Person” with fields “name” and “age” and a method “display()” that prints the name and age of the person. Create a subclass “Employee” that extends “Person” and adds a field “salary” and a method “display()” that prints the name, age, and salary of the employee. Create a subclass “Manager” that extends “Employee” and adds a field “department” and a method “display()” that prints the name, age, salary, and department of the manager. Create an object of the “Person” class and call the “display()” method. Create an object of the “Employee” class and call the “display()” method. Create an object of the “Manager” class and call the “display()” method.

Code:

class Person {

String name;

int age;

Person(String name, int age) {

this.name = name;

this.age = age;

}

void display() {

System.out.println("Name: " + name + ", Age: " + age);

}

}

class Employee extends Person {

double salary;

Employee(String name, int age, double salary) {

super(name, age);

this.salary = salary;

}

@Override

void display() {

System.out.println("Name: " + name + ", Age: " + age + ", Salary: " + salary);

}

}

class Manager extends Employee {

String department;

Manager(String name, int age, double salary, String department) {

super(name, age, salary);

this.department = department;

}

@Override

void display() {

System.out.println("Name: " + name + ", Age: " + age + ", Salary: " + salary + ", Department: " + department);

}

}

public class main7 {

public static void main(String[] args) {

Person p = new Person("Alice", 30);

p.display();

Employee e = new Employee("Bob", 35, 50000);

e.display();

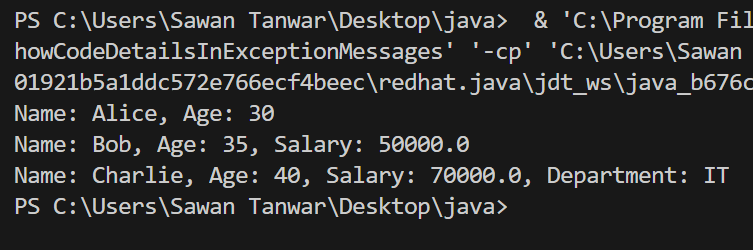
Manager m = new Manager("Charlie", 40, 70000, "IT");

m.display();

}

}

Output:



7. Create a class “Shape” with a method “calculateArea()” that calculates and returns the area of the shape. Create two subclasses “Rectangle” and “Triangle” that extend “Shape” and implement the “calculateArea()” method.

Code:

class Shape {

double calculateArea() {

return 0;

}

}

class Rectangle extends Shape {

double length, width;

Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

@Override

double calculateArea() {

return length \* width;

}

}

class Triangle extends Shape {

double base, height;

Triangle(double base, double height) {

this.base = base;

this.height = height;

}

@Override

double calculateArea() {

return 0.5 \* base \* height;

}

}

public class main7 {

public static void main(String[] args) {

Rectangle r = new Rectangle(5, 10);

System.out.println("Rectangle Area: " + r.calculateArea());

Triangle t = new Triangle(6, 8);

System.out.println("Triangle Area: " + t.calculateArea());

}

}

Output:

