DAY 4

1) Bank_Interest

```
package Day4;

class Bank {
  double getInterestRate() {
    return 0.0;
  }
}

class HDFC extends Bank {
  double getInterestRate() {
    return 8.25;
  }
}

class ICICI extends Bank {
  double getInterestRate() {
    return 8.00;
  }
}

class SBI extends Bank {
  double getInterestRate() {
    return 8.50;
  }
}

public class Interest {
  public static void main(String[] args) {
    Bank b1 = new HDFC();
    Bank b2 = new ICICI();
    Bank b3 = new SBI();
    System.out.println(b1.getInterestRate());
    System.out.println(b2.getInterestRate());
    System.out.println(b3.getInterestRate());
  }
}
```

Output

8.25
8.0
8.5

2) OverloadedCalculator

```
package Day4;

public class OverloadedCalculator {
  int sum(int a, int b)
  {
    return a + b;
  }
  int sum(int a, int b, int c)
  {
    return a + b + c;
  }
  double sum(double a, double b)
  {
```

```
        return a + b;
    }

    public static void main(String[] args) {
        OverloadedCalculator calc = new OverloadedCalculator();
        System.out.println(calc.sum(5, 10));
        System.out.println(calc.sum(5, 10, 15));
        System.out.println(calc.sum(10.5, 20.5));
    }
}
```

Output

15
30
31.0

3) Vehicle

```
package Day4;

class Vehicle {
    void getType() {
        System.out.println("Type of vehicle");
    }
}

class Car extends Vehicle {
    void getType() {
        System.out.println("This is a Car");
    }
}

class Bike extends Vehicle {
    void getType() {
        System.out.println("This is a Bike");
    }
}

public class OverrideExample {
    public static void main(String[] args) {
        Vehicle v1 = new Car();
        v1.getType();
        Vehicle v2 = new Bike();
        v2.getType();
    }
}
```

Output

This is a Car
This is a Bike

4) Electronics

```
package Day4;

class Electronics {
    void showCategory() {
        System.out.println("This is an electronic device.");
    }
}

class Computer extends Electronics {
```

```
    void showDevice() {
        System.out.println("This is a computer.");
    }
}

class Laptop extends Computer {
    void showType() {
        System.out.println("This is a laptop.");
    }
}

class GamingLaptop extends Laptop {
    void showPurpose() {
        System.out.println("This is a gaming laptop.");
    }
}

class Alienware extends GamingLaptop {
    void showBrand() {
        System.out.println("Brand: Alienware");
    }
}

public class MultiInheritance {
    public static void main(String[] args) {
        Alienware a = new Alienware();
        a.showCategory();
        a.showDevice();
        a.showType();
        a.showPurpose();
        a.showBrand();
    }
}
```

Output

This is an electronic device.
This is a computer.
This is a laptop.
This is a gaming laptop.
Brand: Alienware

5) Grandparent

```
package Day4;

class Grandparent {
    int id = 100;
    void info() {
        System.out.println("Grandparent method");
    }
}

class Child extends Grandparent {
    int id = 200;
    void displayId() {
        System.out.println("Child's ID: " + id);
        System.out.println("Grandparent's ID: " + super.id);
        super.info();
    }
}
```

```
public class SuperKeyword {
   public static void main(String[] args) {
      Child c = new Child();
      c.displayId();
   }
}
```

Output

Child's ID: 200

Grandparent's ID: 100

Grandparent method

6) Appliance

```
package Day4;

abstract class Appliance {
   abstract void powerOn();
   abstract void powerOff();
   abstract void runCycle();
}

class Refrigerator extends Appliance {
   void powerOn() { System.out.println("Refrigerator is ON"); }
   void powerOff() { System.out.println("Refrigerator is OFF"); }
   void runCycle() { System.out.println("Cooling food"); }
}

class WashingMachine extends Appliance {
   void powerOn() { System.out.println("Washing Machine is ON"); }
   void powerOff() { System.out.println("Washing Machine is OFF"); }
   void runCycle() { System.out.println("Washing clothes"); }
}

public class Oven extends Appliance {
   void powerOn() { System.out.println("Oven is ON"); }
   void powerOff() { System.out.println("Oven is OFF"); }
   void runCycle() { System.out.println("Baking a cake"); }

   public static void main(String[] args) {
      Appliance[] householdAppliances = {
         new Refrigerator(),
         new WashingMachine(),
         new Oven()
      };

      for (Appliance appliance : householdAppliances) {
         appliance.runCycle();
      }
   }
}
```

Output

Cooling food

Washing clothes

Baking a cake

7) VehicleManufacturing

```
package OOPS;

class VehicleManufacturing {
                                void AssemblyLine() {
                                   System.out.println("Vehicle Assembly Process");
                                   System.out.println("Frame, Engine, Body, Wheels, Interior");
                                }
}

class Car extends VehicleManufacturing {
                                void CarAssembly() {
                                   super.AssemblyLine();
                                   System.out.println("Specifics for Car Assembly");
                                }
}

public class VehicleFactory {
                                public static void main(String[] args) {
                                   Car car = new Car();
                                   car.CarAssembly();
                                }
}
```

Output
Vehicle Assembly Process
Frame, Engine, Body, Wheels, Interior
Specifics for Car Assembly

8) Base&sub

```
package OOPS;

public class base {

                                public base() {
                                   System.out.println("this is superclass constructor");

                                }

}
class sub extends base{

                                public sub() {
                                   super();
                                   System.out.println("This is subclass constructor");
                                }

}
class main1{
                                public static void main(String[] args) {
                                   sub sub = new sub();
                                }
}
```

Output
This is superclass constructor
This is subclass constructor

9) Login

```
package OOPS;
```

```
public class login {

   String username="harshu";
                                private String password;

                                public String getPassword() {
                                   return password;
                                }
   public void setPassword(String password) {
                                   this.password = password;
                                }

   public static void main(String[] args) {
    login l=new login();
    l.setPassword("harshu@175");
    System.out.println("Username: "+ l.username);
    System.out.println("Password: " + l.getPassword());
                                }

}
```

Output :
Username: harshu
Password: harshu@175

10) Multilevel Inheritance

```
package OOPS;

                                class Animal1// parent
                                {
                                   void eat(String eats)
                                   {
                                            System.out.println("Eats="+eats);
                                   }
                                   void sound(String sd)
                                   {
                                            System.out.println("Sound="+sd);
                                   }
                                }
                                class Dog extends Animal1//child
                                {
                                   void lives(String lives)
                                   {
                                            System.out.println("Lives in the "+lives);
                                   }
                                }
                                class Puppy extends Dog//subchild
                                {
                                   void breed(String bd)
                                   {
                                            System.out.println("Breed of Dog is "+bd);
                                   }
                                }

                                public class Multilevel_inheritance {

                                   public static void main(String[] args) {
                                            Puppy max=new Puppy();
```

```
                                        max.eat("Royal canin");
                                        max.sound("barking");
                                        max.lives("city");
                                        max.breed("Lab");

                                    }

                                }
```

Output
Eats=Royal canin
Sound=barking
Lives in the city
Breed of Dog is Lab

11) Career Path

```
package Day4;

class After_12Th {
 void choosePath() {
    System.out.println("Exploring career options after high school.");
 }
}

class Engineering extends After_12Th {
 void program() {
    System.out.println("You have chosen to pursue Engineering.");
 }
}

class IT extends Engineering {
 void specialization() {
    System.out.println("Specialization: Information Technology");
 }
}

class Mechanical extends Engineering {
 void specialization() {
    System.out.println("Specialization: Mechanical Engineering");
 }
}

class CS extends Engineering {
 void specialization() {
    System.out.println("Specialization: Computer Science");
 }
}

class Medical extends After_12Th {
 void program() {
    System.out.println("You have chosen to pursue Medical studies.");
 }
}

class MBBS extends Medical {
 void specialization() {
    System.out.println("Specialization: MBBS");
 }
}
```

```
class BDS extends Medical {
 void specialization() {
    System.out.println("Specialization: BDS");
 }
}

class Other_courses extends After_12Th {
 void program() {
    System.out.println("You have chosen other courses.");
 }
}

class BCA extends Other_courses {
 void specialization() {
    System.out.println("Specialization: Bachelor of Computer Applications");
 }
}

class BBA extends Other_courses {
 void specialization() {
    System.out.println("Specialization: Bachelor of Business Administration");
 }
}

public class TestHierarchy {
 public static void main(String[] args) {
    IT it = new IT();
    it.choosePath();
    it.program();
    it.specialization();

    System.out.println();

    MBBS mbbs = new MBBS();
    mbbs.choosePath();
    mbbs.program();
    mbbs.specialization();

    System.out.println();

    BCA bca = new BCA();
    bca.choosePath();
    bca.program();
    bca.specialization();
 } }
```

Output

Exploring career options after high school.
You have chosen to pursue Engineering.
Specialization: Information Technology

Exploring career options after high school.
You have chosen to pursue Medical studies.
Specialization: MBBS

Exploring career options after high school.
You have chosen other courses.
Specialization: Bachelor of Computer Applications

12) Calculator

```
package Day4;

class Calculator {

  int add(int a, int b) {
    return a + b;
  }

  int add(int a, int b, int c) {
    return a + b + c;
  }

  double add(double a, double b) {
    return a + b;
  }
}

public class TestCalculator {
  public static void main(String[] args) {
    Calculator calc = new Calculator();

    System.out.println("Sum of 2 ints: " + calc.add(10, 20));
    System.out.println("Sum of 3 ints: " + calc.add(5, 15, 8));
    System.out.println("Sum of 2 doubles: " + calc.add(12.3, 8.7));
  }
}
```

Output

Sum of 2 ints: 30
Sum of 3 ints: 28
Sum of 2 doubles: 21.0

13) Smart_Device

```
package Day4;

abstract class MediaPlayer {
  abstract void start();
  abstract void stop();
  abstract void playMedia();
}

class MP3Player extends MediaPlayer {
  @Override
  void start() {
    System.out.println("MP3 Player is starting up.");
  }

  @Override
  void stop() {
    System.out.println("MP3 Player is shutting down.");
  }

  @Override
  void playMedia() {
    System.out.println("Playing audio file...");
  }
}
```

```
class VideoPlayer extends MediaPlayer {
    @Override
    void start() {
        System.out.println("Video Player is starting up.");
    }

    @Override
    void stop() {
        System.out.println("Video Player is shutting down.");
    }

    @Override
    void playMedia() {
        System.out.println("Playing video file...");
    }
}

class StreamingService extends MediaPlayer {
    @Override
    void start() {
        System.out.println("Streaming Service is connecting.");
    }

    @Override
    void stop() {
        System.out.println("Streaming Service is disconnecting.");
    }

    @Override
    void playMedia() {
        System.out.println("Streaming content from the internet...");
    }
}

// Main class to test
public class TestPlayers {
    public static void main(String[] args) {
        MediaPlayer[] players = {
            new MP3Player(),
            new VideoPlayer(),
            new StreamingService()
        };

        for (MediaPlayer player : players) {
            player.start();
            player.playMedia();
            player.stop();
            System.out.println("--------------------");
        }
    }
}
```

Output

MP3 Player is starting up.
Playing audio file...
MP3 Player is shutting down.

Video Player is starting up.
Playing video file...

Video Player is shutting down.

Streaming Service is connecting.
Streaming content from the internet...
Streaming Service is disconnecting.

14) Hospital

```
package Day4;

class Clinic {
   String clinicName = "GM Hospital";
   String address = "BasaveshwaraNagar, Bangalore Karnataka ";

   void clinicDetails() {
      System.out.println("Clinic Name: " + clinicName);
      System.out.println("Address: " + address);
   }
}

class Client extends Clinic {
   String clientName;
   int clientAge;

   Client(String clientName, int clientAge) {
      this.clientName = clientName;
      this.clientAge = clientAge;
   }

   void clientDetails() {
      System.out.println("Client Name: " + clientName);
      System.out.println("Age: " + clientAge);
      System.out.println("--- Clinic Information ---");
      clinicDetails();
   }
}

public class TestClinic {
   public static void main(String[] args) {
      Client c1 = new Client("Harshu", 23);
      c1.clientDetails();
   }
}
```

Output
Client Name: Harshu
Age: 23
--- Clinic Information ---
Clinic Name: GM Hospital
Address: BasaveshwaraNagar, Bangalore Karnataka

15) Employee

```
package Day4;

class Employee {
   private String name;
   private int employeeId;
   private double salary;

   public Employee(String name, int employeeId, double salary) {
```

```
        this.name = name;
        this.employeeId = employeeId;
        if (salary >= 0) {
            this.salary = salary;
        } else {
            this.salary = 0.0;
        }
    }

    public String getName() {
        return name;
    }

    public int getEmployeeId() {
        return employeeId;
    }

    public double getSalary() {
        return salary;
    }

    public void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Employee ID: " + employeeId);
        System.out.println("Salary: " + salary);
    }

    public void updateSalary(double newSalary) {
        if (newSalary > this.salary) {
            this.salary = newSalary;
        } else {
            System.out.println("Invalid or lower salary. Update rejected.");
        }
    }
}

public class EmployeeManagement {
    public static void main(String[] args) {
        Employee emp1 = new Employee("Harshu", 205, 55000.0);
        emp1.displayInfo();

        emp1.updateSalary(50000.0);

        emp1.updateSalary(60000.0);

        System.out.println("After attempting update:");
        emp1.displayInfo();
    }
}
```

Output

Name: Harshu
Employee ID: 205
Salary: 55000.0
Invalid or lower salary. Update rejected.
After attempting update:
Name: Harshu
Employee ID: 205
Salary: 60000.0