AIM: Write a java program to implement JDK 8 Features

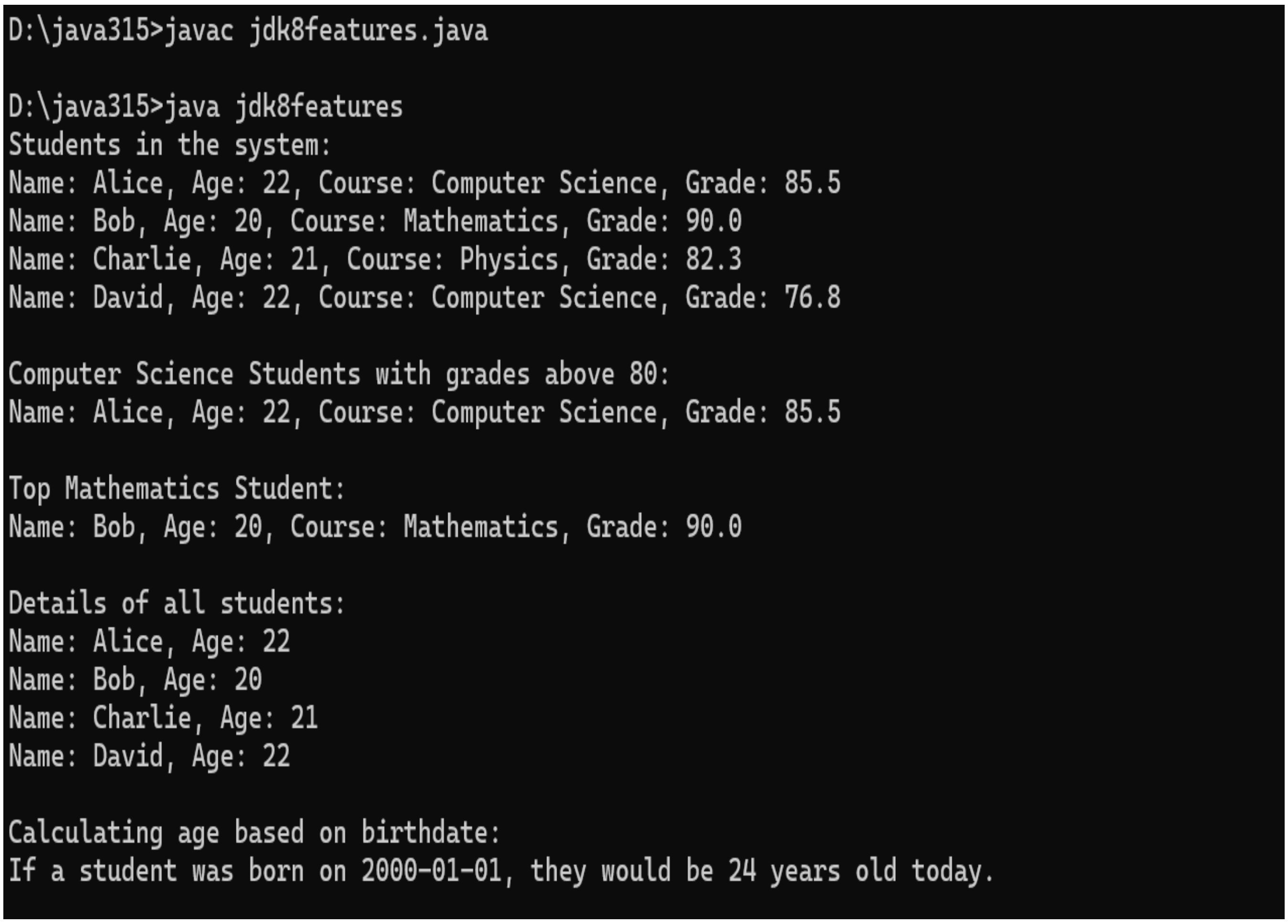
PROGRAM:
```
import java.time.*;
import java.util.*;
import java.util.function.Predicate;
import java.util.stream.Collectors;
interface Person {
    String getName();
    int getAge();

    default String getPersonInfo() {
        return "Name: " + getName() + ", Age: " + getAge();
    }

    static void printPerson(Person person) {
        System.out.println(person.getPersonInfo());
    }
}
class Student implements Person {
    private String name;
    private int age;
    private String course;
    private double grade;
    public Student(String name, int age, String course, double grade) {
        this.name = name;
        this.age = age;
        this.course = course;
        this.grade = grade;
    }
    public String getName() { return name; }
    public int getAge() { return age; }
    public String getCourse() { return course; }
    public double getGrade() { return grade; }
    public String toString() {
        return getPersonInfo() + ", Course: " + course + ", Grade: " + grade;
    }
}

public class jdk8features {
    public static void main(String[] args) {
        List<Student> students = Arrays.asList(
            new Student("Alice", 22, "Computer Science", 85.5),
            new Student("Bob", 20, "Mathematics", 90.0),
            new Student("Charlie", 21, "Physics", 82.3),
            new Student("David", 22, "Computer Science", 76.8)
        );
        System.out.println("Students in the system:");
        students.forEach(System.out::println); // Method reference
        System.out.println("\nComputer Science Students with grades above 80:");
        List<Student> csStudents = students.stream()
            .filter(s -> s.getCourse().equals("Computer Science") && s.getGrade() > 80)
            .collect(Collectors.toList());
        csStudents.forEach(System.out::println);
        Optional<Student> topMathStudent = students.stream()
            .filter(s -> s.getCourse().equals("Mathematics"))
            .findFirst();
        System.out.println("\nTop Mathematics Student:");
        System.out.println(topMathStudent.orElse(new Student("No Student", 0,
"None", 0.0)));
        System.out.println("\nDetails of all students:");
        students.forEach(Person::printPerson);
        System.out.println("\nCalculating age based on birthdate:");
        LocalDate birthDate = LocalDate.of(2000, Month.JANUARY, 1);
        LocalDate currentDate = LocalDate.now();
        Period age = Period.between(birthDate, currentDate);
        System.out.println("If a student was born on " + birthDate + ", they would be " +
age.getYears() + " years old today.");
    }
}
```

OUTPUT:



```
D:\java315>javac jdk8features.java

D:\java315>java jdk8features
Students in the system:
Name: Alice, Age: 22, Course: Computer Science, Grade: 85.5
Name: Bob, Age: 20, Course: Mathematics, Grade: 90.0
Name: Charlie, Age: 21, Course: Physics, Grade: 82.3
Name: David, Age: 22, Course: Computer Science, Grade: 76.8

Computer Science Students with grades above 80:
Name: Alice, Age: 22, Course: Computer Science, Grade: 85.5

Top Mathematics Student:
Name: Bob, Age: 20, Course: Mathematics, Grade: 90.0

Details of all students:
Name: Alice, Age: 22
Name: Bob, Age: 20
Name: Charlie, Age: 21
Name: David, Age: 22

Calculating age based on birthdate:
If a student was born on 2000-01-01, they would be 24 years old today.
```