Visual and Net Based Programming (Theory+Lab)

INTRODUCTION

Agnik Saha

Department of Computer Science and Engineering

R. P. Shaha University

August 22, 2023

Course Timings

- **Theory**
  - **Saturday -> (9:00 am - 10:20 am)**
  - **Tuesday -> (11:50 am - 1:00 pm)**

- **Lab**
  - **Tuesday -> (2:10 pm - 3:20 pm)**

Course Logistics

CSE 225

- 2 Class Tests (15%)
- 2 Assignments (20%)
- Class Participation (5%)
- Mid Term (20%)
- Final Exam (40%)

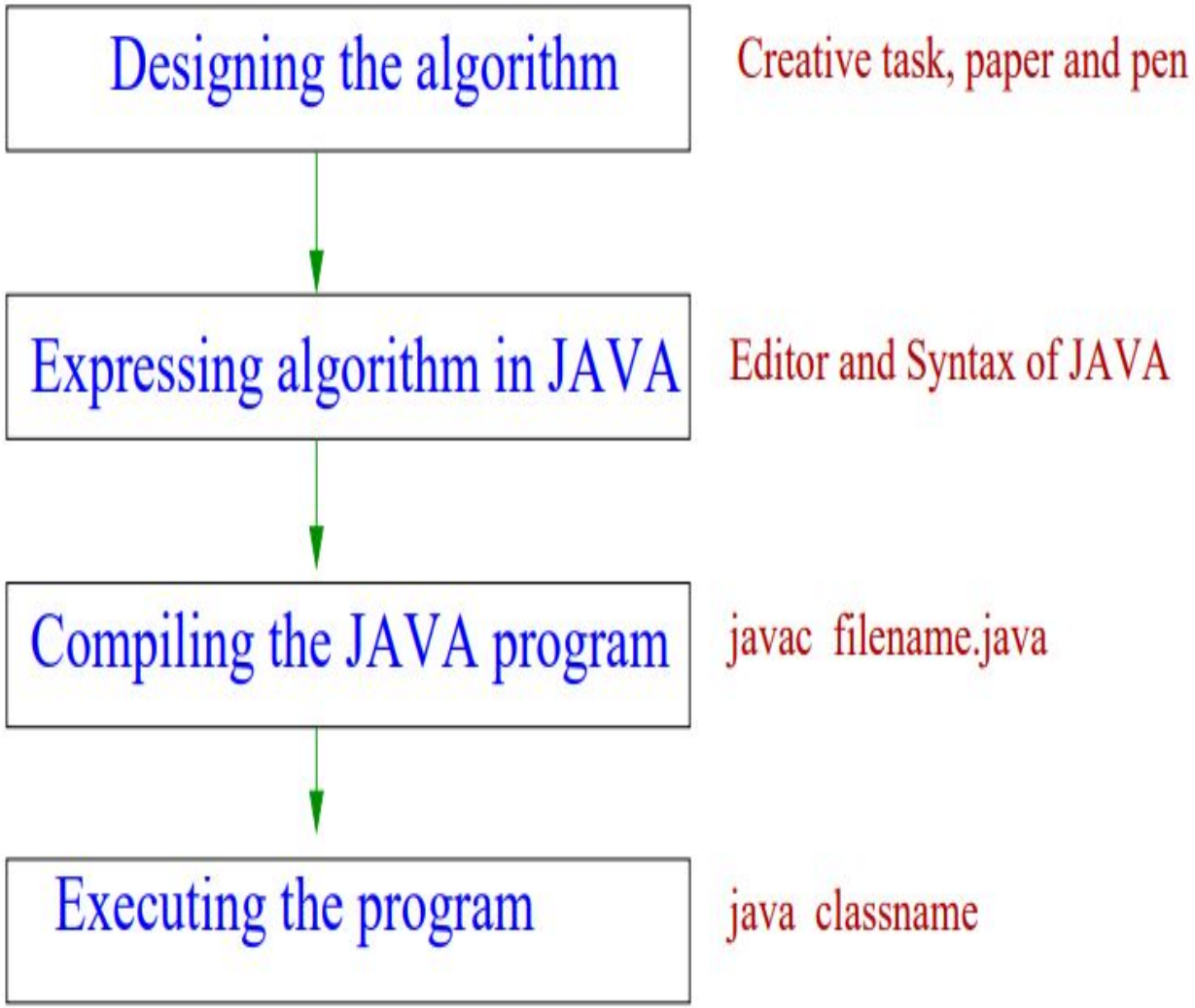
CSE 226

- 6 Assignments (40%)
- Final Assessments (60%)
  - Viva (10%)
  - Final Exam (50%)

What will we gain/learn from the course ?

- **Develop algorithmic skills**
- **How to use computers**
- **Expressing algorithm as programs in JAVA : Editor**
- **How to make these JAVA programs run on a computer : compiling, executing**
- **From Basics to Advanced Course**

Steps in Solving the problem



Java Installation

1.

```
java -version
```

```
C:\Users\boskom>java -version
'java' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\boskom>
```

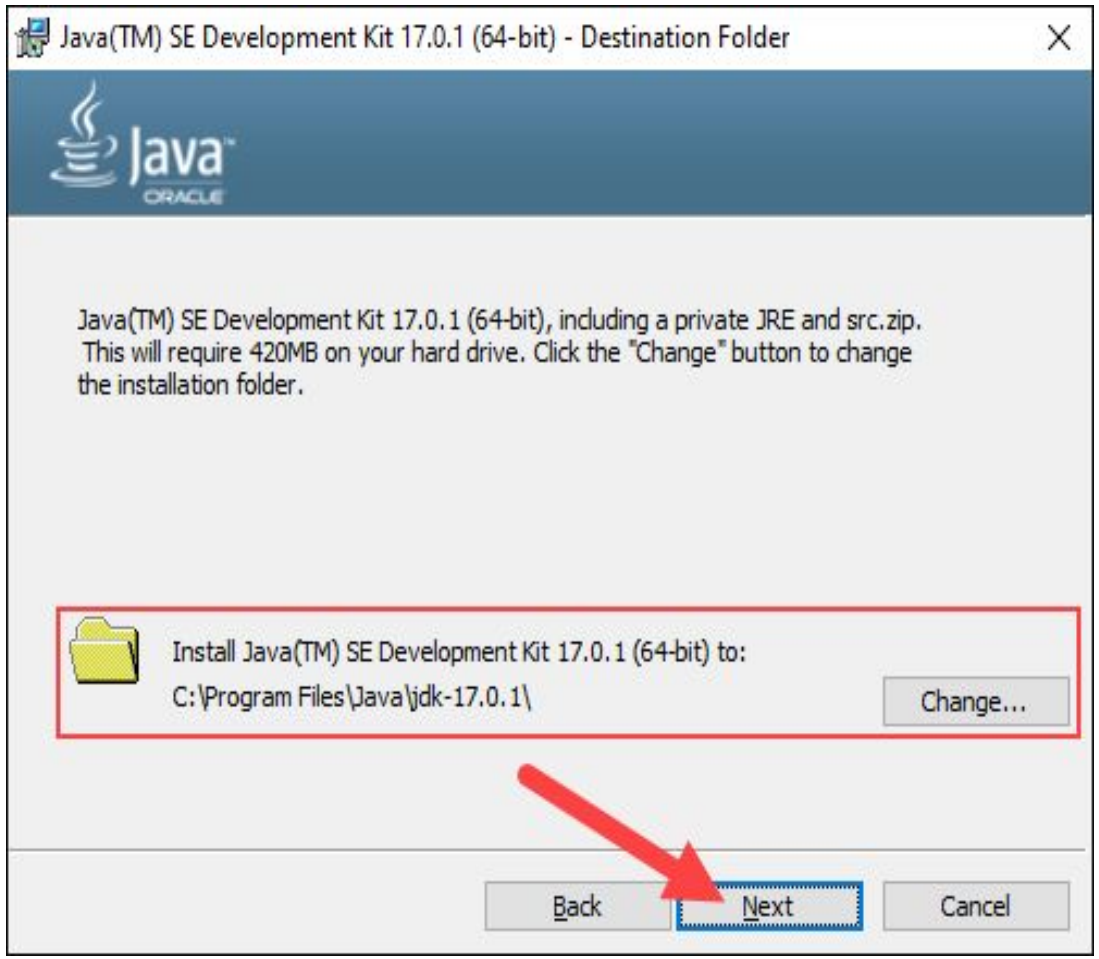
2. Visit the website: https://www.oracle.com/java/technologies/downloads/#jdk17-windows

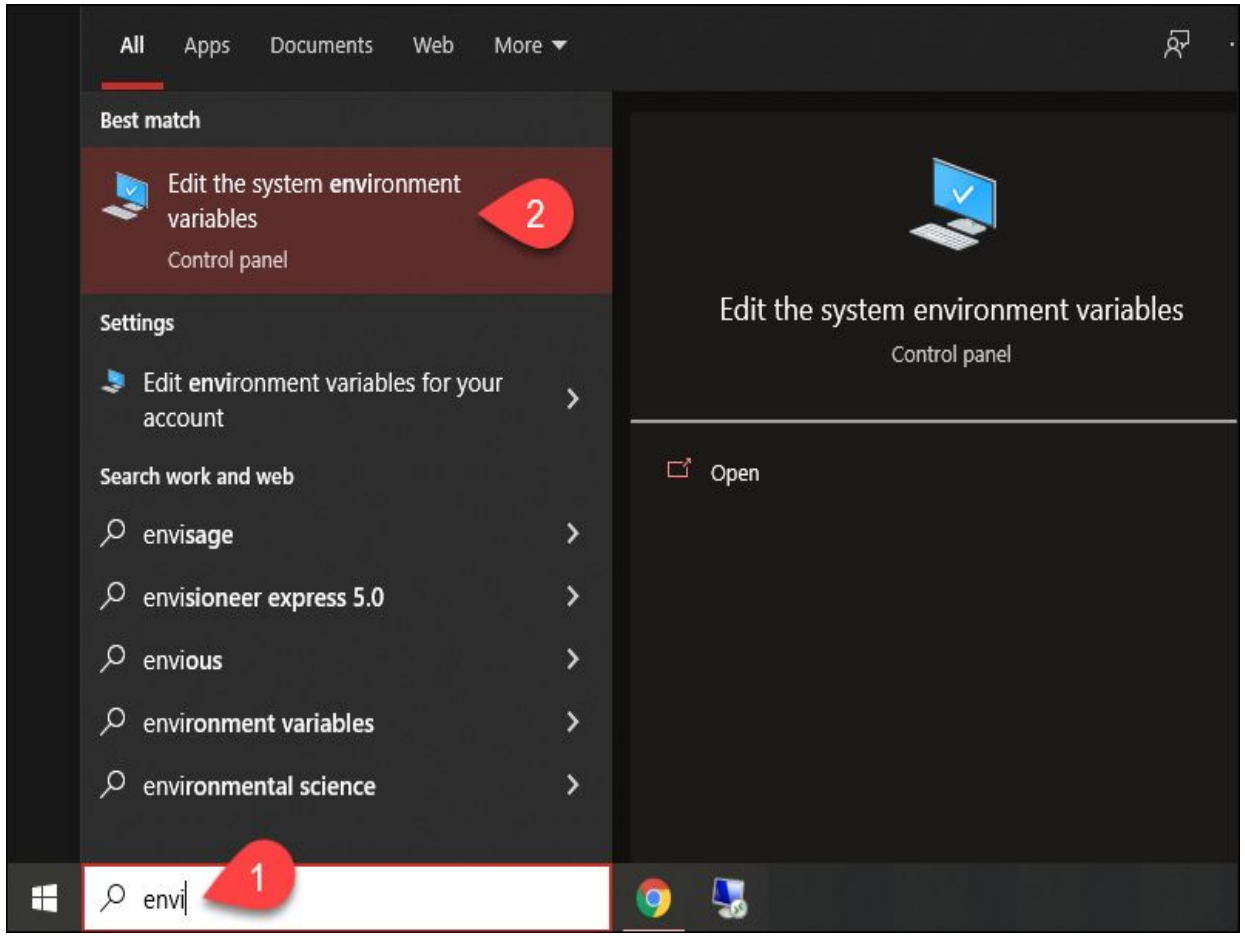
3.

Linux macOS Windows

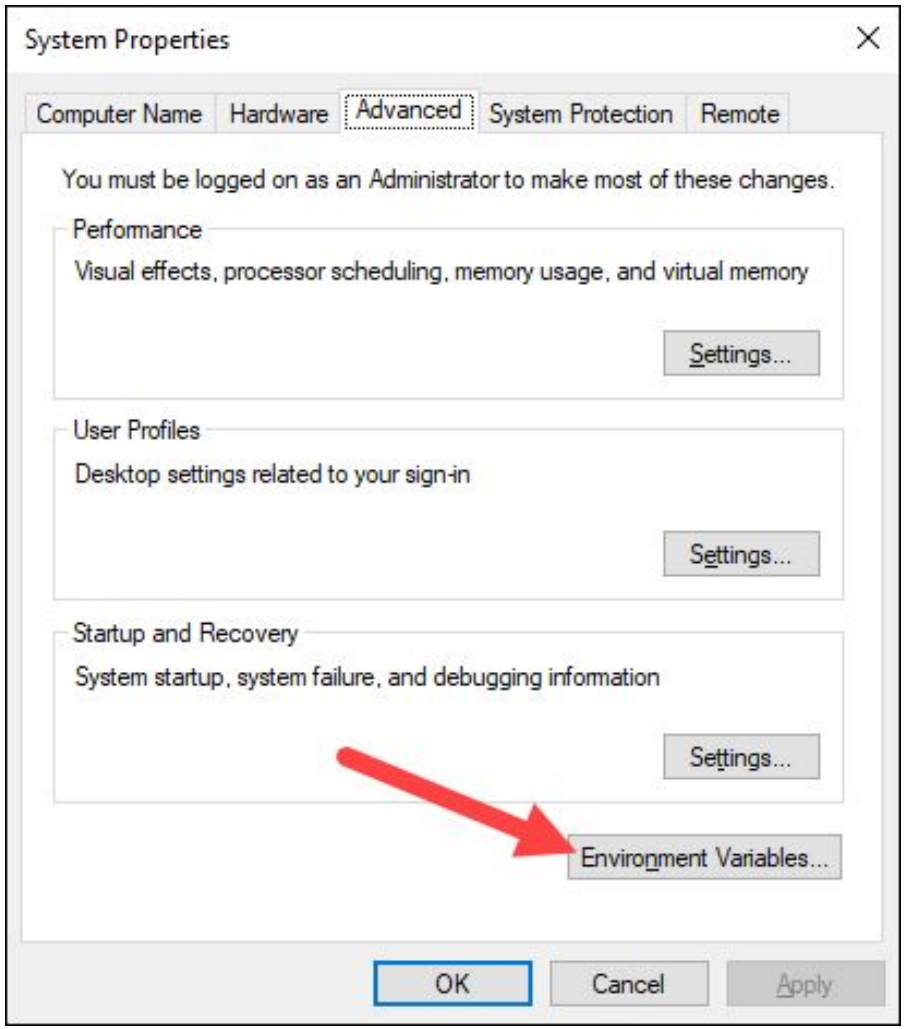
Product/file description	File size	Download
x64 Compressed Archive	170.66 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256)
x64 Installer	152 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256)
x64 MSI Installer	150.89 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256)

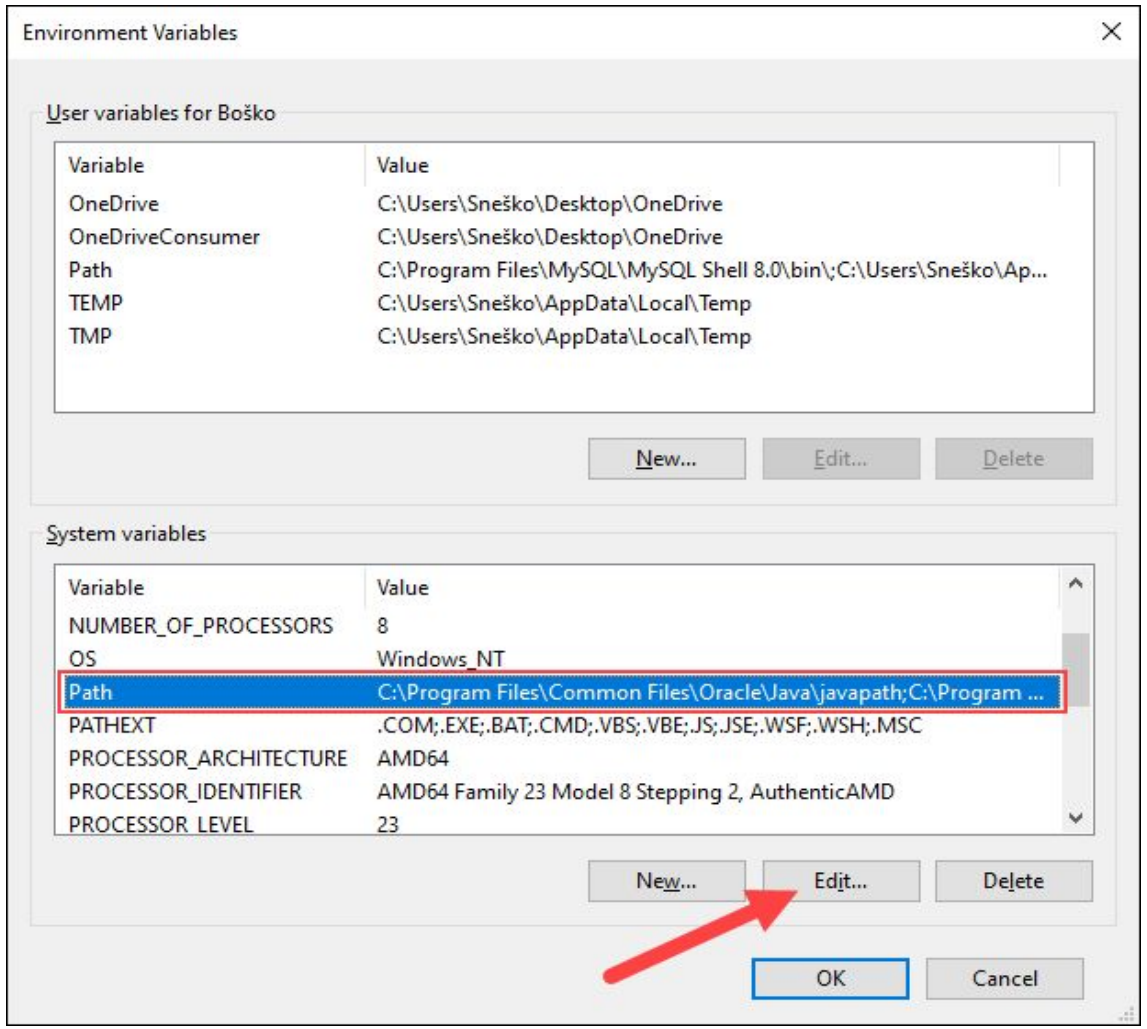
Java Installation



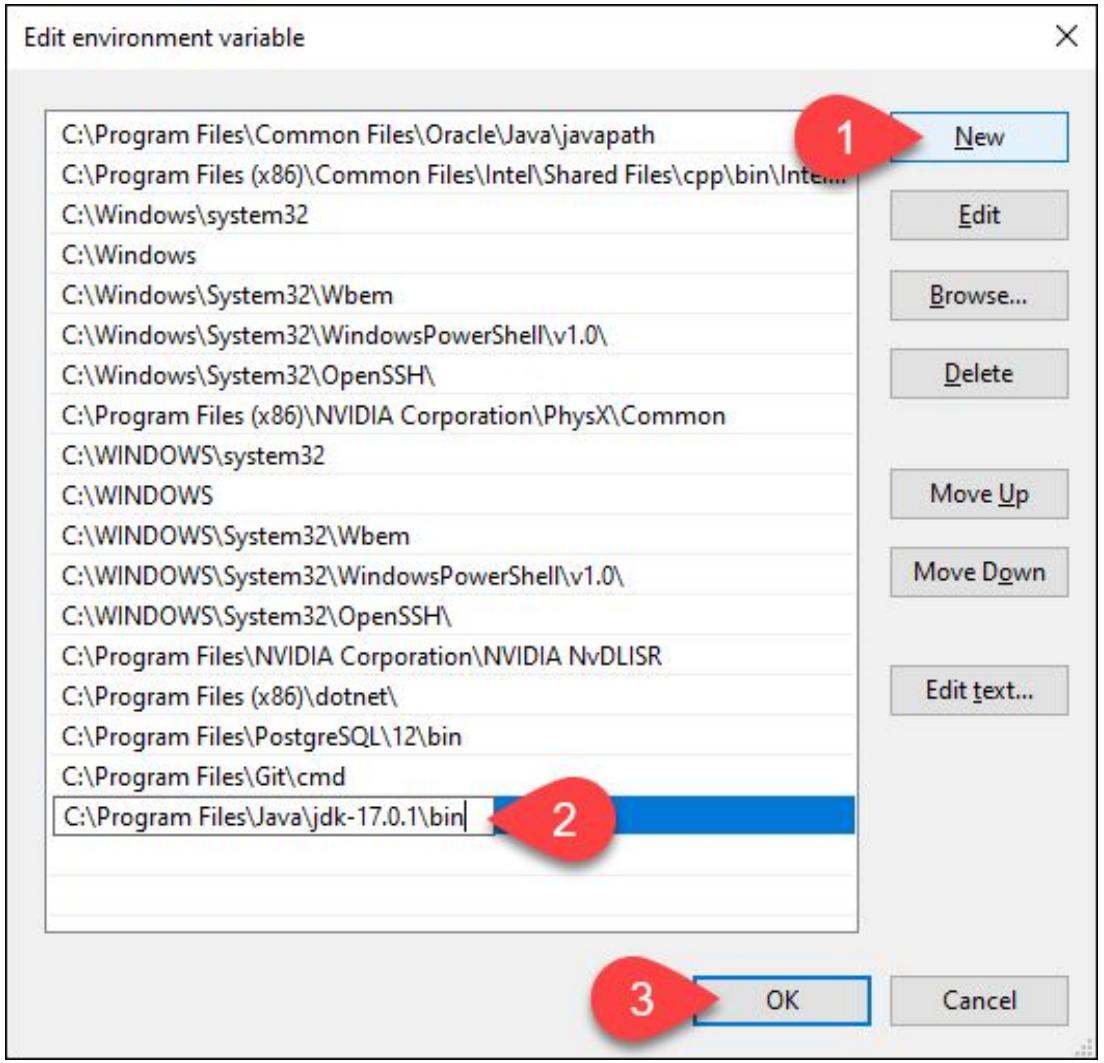


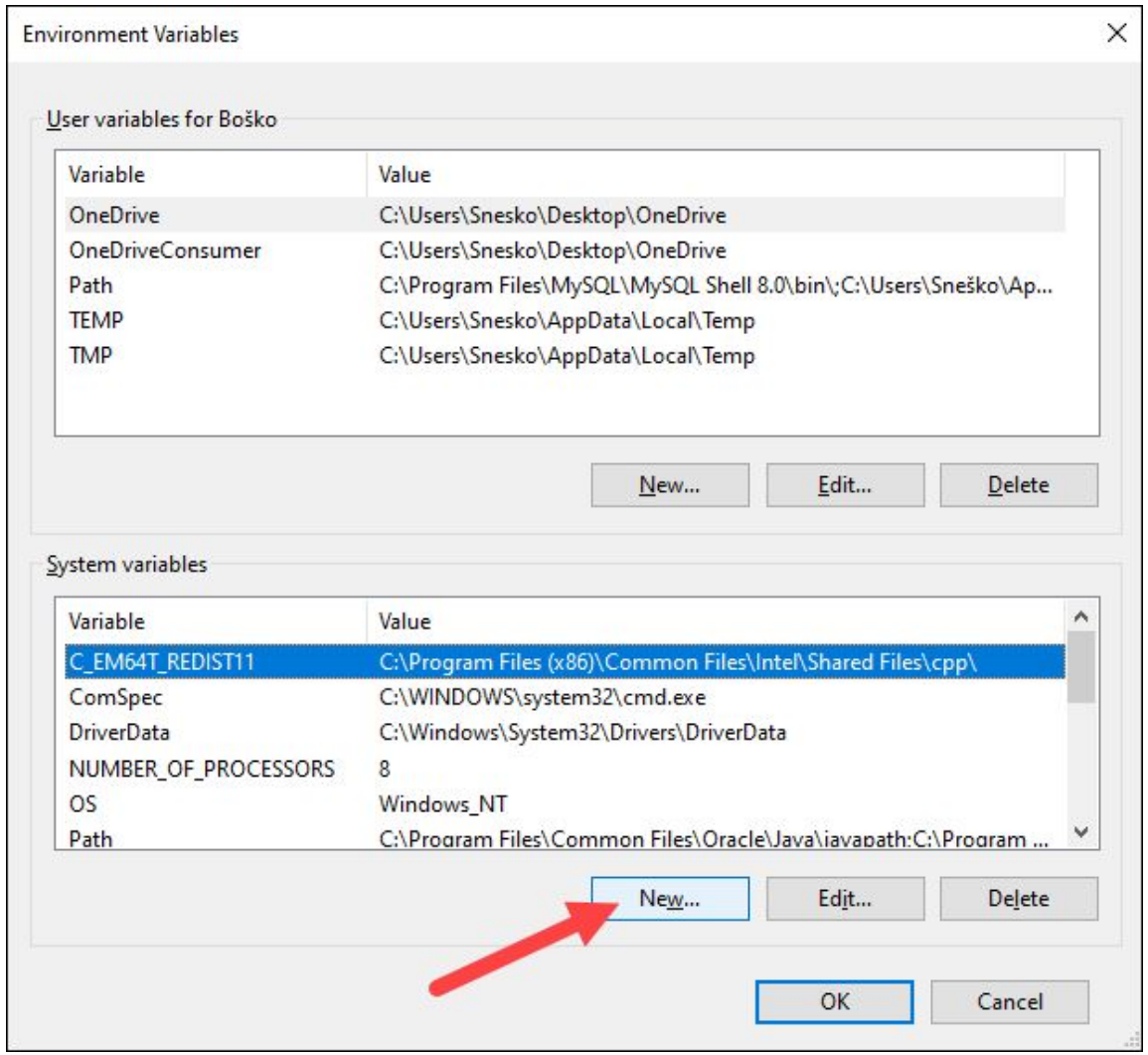
Java Installation



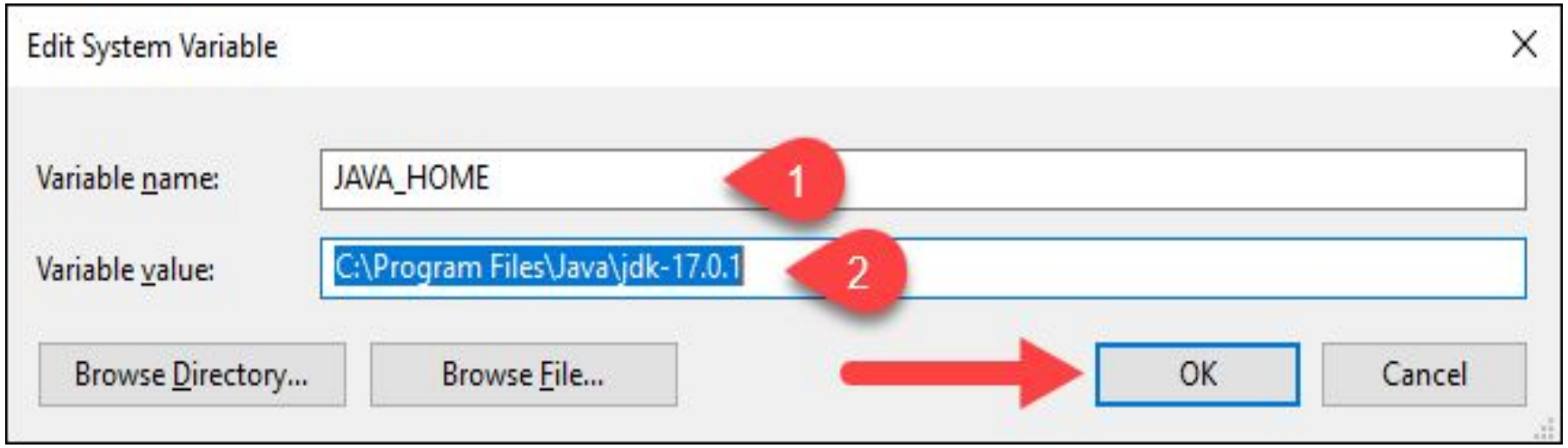


Java Installation





Java Installation




```
C:\Users\boskom>java -version
java version "17.0.1" 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)
```

Building a student ERP system

Perform the following operations:

- **Store the First name of the student.**
- **Store the Last name of the student.**
- **Store the unique Roll number for every student.**
- **Store the CGPA of every student.**
- **Store the courses registered by the student.**
- **Store the course-wise marks of the student**

Building a student ERP system

- **Add Student Details: Get data from user and add a student to the list of students. While adding the students into the list, check for the uniqueness of the roll number.**

- **Find the student by the given roll number: This function is to find the student record for the given roll number and print the details.**

Building a student ERP system

- **Find the student by the given first name: This function is to find all the students with the given first name and print their details.**

- **Find the students registered in a course: This function is to find all the students who have registered for a given course.**

- **Count of Students: This function is to print the total number of students in the system**

Building a student ERP system

- **Delete a student: This function is to delete the student record for the given roll number.**

- **Update Student: This function is to update the student records. This function does not ask for new details for all fields but the user should be able to pick and choose what he wants to update.**

- **Get the top 10 students based on cgpa**

Let’s Code together….

- Java utils -> ArrayList, HashMap, List, Map, Scanner

```java
class Student {
    String firstName;
    String lastName;
    String rollNumber;
    double cgpa;
    List<String> courses;
    Map<String, Integer> courseMarks;

    public Student(String firstName, String lastName, String rollNumber, double cgpa) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.rollNumber = rollNumber;
        this.cgpa = cgpa;
        this.courses = new ArrayList<>();
        this.courseMarks = new HashMap<>();
    }
}
```

Let’s Code together….

- private static List<Student> students = new ArrayList<>();

```
private static void addStudent(Scanner scanner) {
    System.out.print("Enter First Name: ");
    String firstName = scanner.nextLine();

    System.out.print("Enter Last Name: ");
    String lastName = scanner.nextLine();

    System.out.print("Enter Roll Number: ");
    String rollNumber = scanner.nextLine();

    // Check for uniqueness of roll number
    boolean isUniqueRollNumber = students.stream().noneMatch(student -> student.rollNumber.equals(rollNumber));
    if (!isUniqueRollNumber) {
        System.out.println("Roll number is already in use. Please enter a unique roll number.");
        return;
    }

    System.out.print("Enter CGPA: ");
    double cgpa = scanner.nextDouble();
    scanner.nextLine(); // Consume newline

    Student student = new Student(firstName, lastName, rollNumber, cgpa);
```

Let’s Code together….

```
        System.out.print("Enter number of courses registered: ");
        int numCourses = scanner.nextInt();
        scanner.nextLine(); // Consume newline
        for (int i = 0; i < numCourses; i++) {
            System.out.print("Enter course name: ");
            String courseName = scanner.nextLine();
            student.courses.add(courseName);

            System.out.print("Enter marks for course " + courseName + ": ");
            int marks = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            student.courseMarks.put(courseName, marks);
        }

        students.add(student);
        System.out.println("Student added successfully!");
    }
```

Let’s Code together….

```
private static void findStudentByRollNumber(Scanner scanner) {
    System.out.print("Enter Roll Number to search: ");
    String searchRollNumber = scanner.nextLine();

    for (Student student : students) {
        if (student.rollNumber.equals(searchRollNumber)) {
            System.out.println("Student Found:");
            System.out.println("First Name: " + student.firstName);
            System.out.println("Last Name: " + student.lastName);
            System.out.println("Roll Number: " + student.rollNumber);
            System.out.println("CGPA: " + student.cgpa);
            System.out.println("Courses Registered: " + student.courses);
            System.out.println("Course-wise Marks: " + student.courseMarks);
            return;
        }
    }

    System.out.println("Student with Roll Number " + searchRollNumber + " not found.");
}
```

Let’s Code together….

```
        int choice = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        switch (choice) {
            case 1:
                addStudent(scanner);
                break;
            case 2:
                findStudentByRollNumber(scanner);
                break;
            case 3:
                findStudentsByFirstName(scanner);
                break;
            case 4:
                findStudentsByCourse(scanner);
                break;
            case 5:
                countStudents();
                break;
            case 6:
                deleteStudent(scanner);
                break;
            case 7:
                updateStudent(scanner);
                break;
            case 8:
                getTopStudentsByCGPA();
                break;
            case 9:
                System.out.println("Exiting...");
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice. Please select a valid option.");
        }
    }
}
```

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

THANK YOU