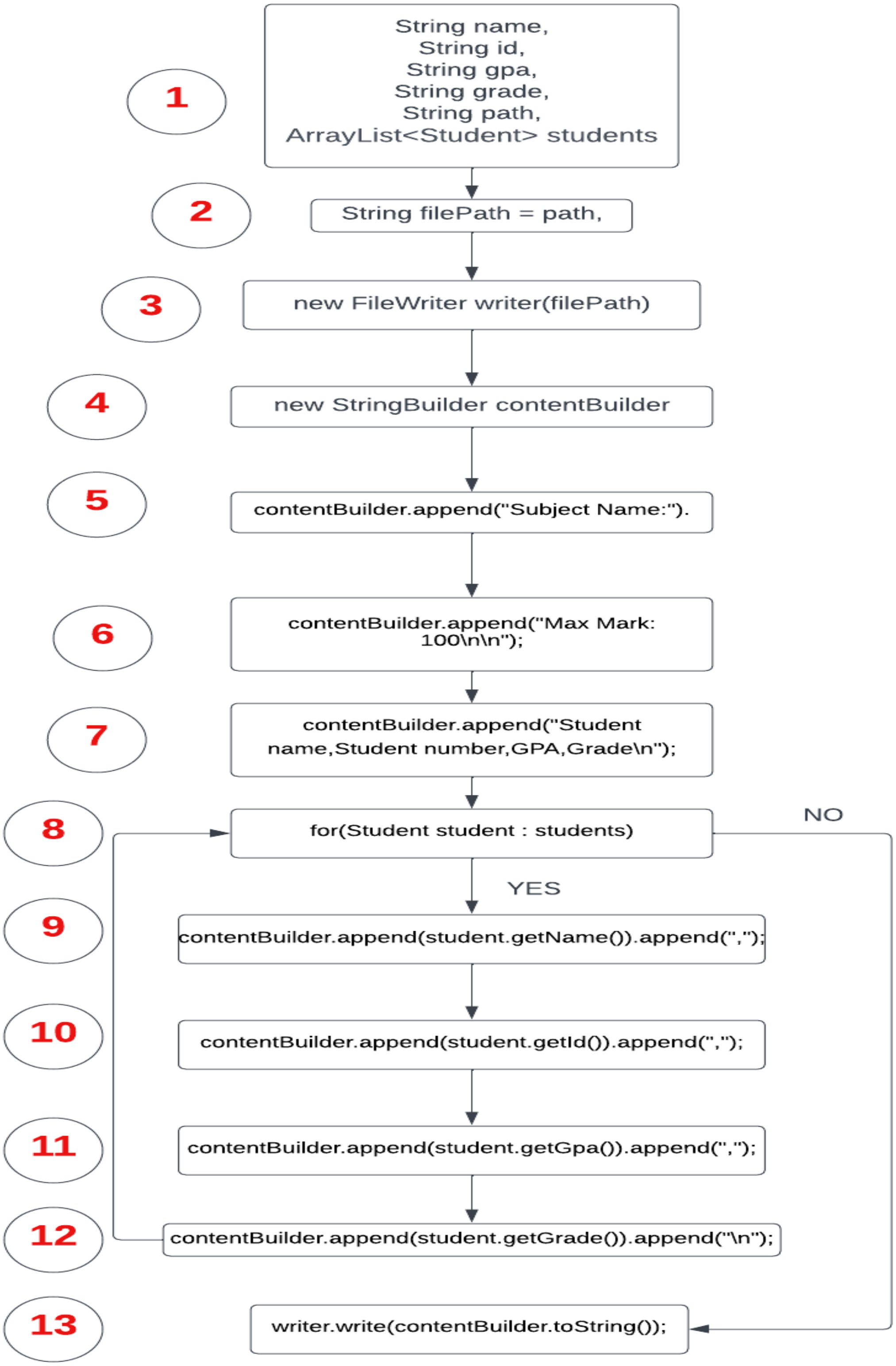
# Data Flow Coverage

## writeFile() function Code:

```
1- //Declaration of student name, id, gpa, grade, path;

public void write_file() throws IOException  {

2-  String filePath = path.substring(0, path.lastIndexOf('/') + 1) + "Results.csv";

3-   try (FileWriter writer = new FileWriter(filePath)) {

4-     StringBuilder contentBuilder = new StringBuilder();

5-     contentBuilder.append("Subject Name:").append(course.getName()).append(",,,,,");

6-    contentBuilder.append("Max Mark: 100\n\n");

7-   contentBuilder.append("Student name,Student number,GPA,Grade\n");

8-     for (Student student : students) {

9-        contentBuilder.append(student.getName()).append(",");

10-       contentBuilder.append(student.getId()).append(",");

11-     contentBuilder.append(student.getGpa()).append(",");

12-    contentBuilder.append(student.getGrade()).append("\n");  }

13-   writer.write(contentBuilder.toString());

14-  System.out.println("Successfully wrote to the file.");

      }

   }
```

# Control Flow Graph:


1
String name,
String id,
String gpa,
String grade,
String path,
ArrayList<Student> students
2
String filePath = path,
3
new FileWriter writer(filePath)
4
new StringBuilder contentBuilder
5
contentBuilder.append("Subject Name:").
6
contentBuilder.append("Max Mark: 100\n\n");
7
contentBuilder.append("Student name,Student number,GPA,Grade\n");
8
for(Student student : students)
NO
YES
9
contentBuilder.append(student.getName()).append(",");
10
contentBuilder.append(student.getId()).append(",");
11
contentBuilder.append(student.getGpa()).append(",");
12
contentBuilder.append(student.getGrade()).append("\n");
13
writer.write(contentBuilder.toString());

# Defined-Used Variables:

| Line | Define | Use |
|---|---|---|
| 1 | name, id, gpa, grade, path, students | |
| 2 | filePath | path |
| 3 | writer | filePath |
| 4 | contentBuilder | |
| 5 | | contentBuilder |
| 6 | | contentBuilder |
| 7 | | contentBuilder |
| 8 | student | students |
| 9 | | contentBuilder, name, student |
| 10 | | contentBuilder, id, student |
| 11 | | contentBuilder, gpa, student |
| 12 | | contentBuilder, grade, student |
| 13 | | contentBuilder, writer |

# DU-Pairs and Paths for all variables:

## For the variable name:

| DU-Pair | Path |
|---|---|
| (1, 9) | <1, 2, 3, 4, 5, 6, 7, 8, 9> |

## For the variable id:

| DU-Pair | Path |
|---|---|
| (1, 10) | <1, 2, 3, 4, 5, 6, 7, 8, 9, 10> |

## For the variable gpa:

| DU-Pair | Path |
|---|---|
| (1, 11) | <1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11> |

## For the variable grade:

| DU-Pair | Path |
|---|---|
| (1, 12) | <1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12> |

## For the variable path:

| DU-Pair | Path |
|---|---|
| (1, 2) | <1, 2> |

## For the variable students:

| DU-Pair | Path |
|---|---|
| (1, <8, 9>) | <1, 2, 3, 4, 5, 6, 7, 8, 9> |
| (1, <8, 13>) | <1, 2, 3, 4, 5, 6, 7, 8, 13> |

### For the variable filePath:

| DU-Pair | Path |
|---|---|
| (2, 3) | <2, 3> |

### For the variable writer:

| DU-Pair | Path |
|---|---|
| (3, 13) | <3, 4, 5, 6, 7, 8, 13> |
|  | <3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13> |

### For the variable contentBuilder:

| DU-Pair | Path |
|---|---|
| (4, 5) | <4, 5> |
| (4, 6) | <4, 5, 6> |
| (4, 7) | <4, 5, 6, 7> |
| (4, 9) | <4, 5, 6, 7, 8, 9> |
| (4, 10) | <4, 5, 6, 7, 8, 9, 10> |
| (4, 11) | <4, 5, 6, 7, 8, 9, 10, 11> |
| (4, 12) | <4, 5, 6, 7, 8, 9, 10, 11, 12> |
| (4, 13) | <4, 5, 6, 7, 8, 9, 10, 11, 12, 13> |
|  | <4, 5, 6, 7, 8, 13> |

### For the variable student:

| DU-Pair | Path |
|---|---|
| (8, 9) | <8, 9> |
| (8, 10) | <8, 9, 10> |
| (8, 11) | <8, 9, 10, 11> |
| (8, 12) | <8, 9, 10, 11, 12> |

## All-Du coverage:

**- Apply the test case on path <1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13> to achieve All-Du coverage, so that each definition for each variable reaches each use for each of them.**

**- Test case:**

**path = “.\\src\\test\\java\\InputFile TestCases\\fileName.txt”**

**students = {**

**name = “Alice Dodo”,**

**id = “15367901”,**

**activitiesMark = 5,**

**oralPracticalMark = 5,**

**midtermMark = 6,**

**finalMark = 60,**

**totalMark = 76,**

**grade = “A+”,**

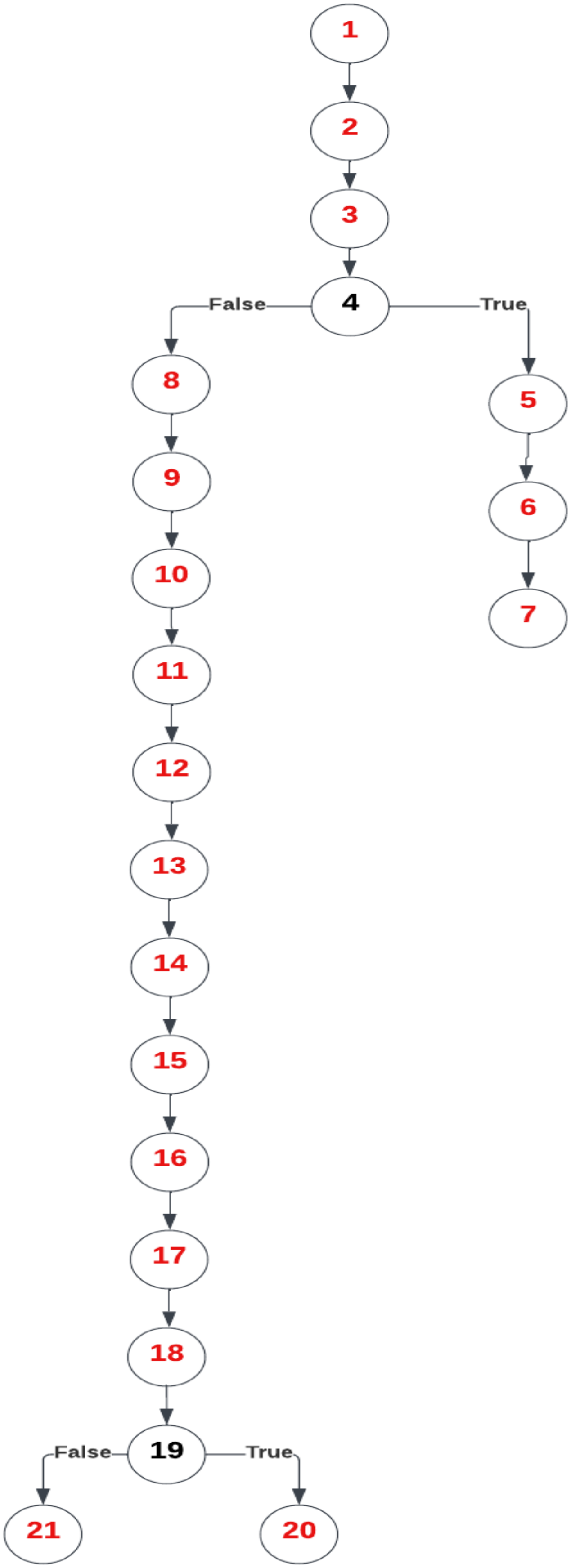
**gpa = 4**

**}**

# store_data () function Code:

```
1- private final ArrayList<Student> students; private final Course course;

2 - public void store_data(String line, int lineNumber) throws
InvalidCourseNameException,InvalidCourseCodeException, InvalidStudentMarksException,
InvalidStudentIdException, InvalidStudentNameException, InvalidCourseMarkException {
 3- String[] words =line.split(",");
 4- if(lineNumber == 0)   {
 5-      course.setName(words[0]);
 6-    course.setCode(words[1]);
 7-      course.setMaxMark(words[2]);}
   else {
   8-   Student student = new Student();
   9-    student.setName(words[0]);
    10-   student.setId(words[1]);
   11-  student.setActivitiesMark(Integer.parseInt(words[2]));
   12-  student.setOralPracticalMark(Integer.parseInt(words[3]));
   13-    student.setMidtermMark(Integer.parseInt(words[4]));
   14-  student.setFinalMark(Integer.parseInt(words[5]));
   15- student.setTotalMark(Calculator.compute_total_mark(student.getActivitiesMark(),

          student.getOralPracticalMark(),student.getMidtermMark(),student.getFinalMark()));
  16-    student.setGrade(Calculator.compute_Grade(student.getTotalMark()));
  17-   student.setGpa(Calculator.compute_GPA(student.getGrade()));
  18-    boolean hasInvalidId = students.stream().anyMatch(n -> n.getId().equals(words[1]));
  19-    if (hasInvalidId) {
  20-        throw new InvalidStudentIdException(InvalidStudentIdException.
INVALID_STUDENT_ID_REPEATED);}
  21-    students.add(student); }}
```

# Control Flow Graph:



## Defined-Used Variables:

| Line | Define | Use |
|---|---|---|
| 1 | students, course | |
| 2 | line, lineNumber | |
| 3 | words | line |
| 4 | | lineNumber |
| 5 | | Course, words |
| 6 | | Course, words |
| 7 | | Course, words |
| 8 | student | |
| 9 | | student, words |
| 10 | | student, words |
| 11 | | student, words |
| 12 | | student, words |
| 13 | | student, words |
| 14 | | student, words |
| 15 | | student |
| 16 | | student |
| 17 | | student |
| 18 | hasInvalidId | students |
| 19 | | hasInvalidId |
| 20 | | |
| 21 | | students, student |

## For the variable students:

| DU-Pair | Path |
|---|---|
| (1, 18) | <1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18> |
| (1, 21) | <1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21> |

## For the variable course:

| DU-Pair | Path |
|---|---|
| (1, 5) | <1, 2, 3, 4, 5> |
| (1, 6) | <1, 2, 3, 4, 5, 6> |
| (1, 7) | <1, 2, 3, 4, 5, 6, 7> |

## For the variable line:

| DU-Pair | Path |
|---|---|
| (2, 3) | <2, 3> |

## For the variable lineNumber:

| DU-Pair | Path |
|---|---|
| (2, 4) | <2, 3, 4> |

## For the variable words:

| DU-Pair | Path |
|---|---|
| (3, 9) | <3, 4, 8, 9> |
| (3, 10) | <3, 4, 8, 9, 10> |
| (3, 11) | <3, 4, 8, 9, 10, 11> |
| (3, 12) | <3, 4, 8, 9, 10, 11, 12> |
| (3, 13) | <3, 4, 8, 9, 10, 11, 12, 13> |
| (3, 14) | <3, 4, 8, 9, 10, 11, 12, 13, 14> |

### For the variable student:

| DU-Pair | Path |
|---|---|
| (8, 9) | <8, 9> |
| (8, 10) | <8, 9, 10> |
| (8, 11) | <8, 9, 10, 11> |
| (8, 12) | <8, 9, 10, 11, 12> |
| (8, 13) | <8, 9, 10, 11, 12, 13> |
| (8, 14) | <8, 9, 10, 11, 12, 13, 14> |
| (8, 15) | <8, 9, 10, 11, 12, 13, 14, 15> |
| (8, 16) | <8, 9, 10, 11, 12, 13, 14, 15, 16> |
| (8, 17) | <8, 9, 10, 11, 12, 13, 14, 15, 16, 17> |
| (8, 21) | <8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21> |

### For the variable hasInvalidId:

| DU-Pair | Path |
|---|---|
| (18, 19) | <18, 19> |

## All-Du coverage:

**- Two test cases must be applied:**

**<1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21>**

**And <1, 2, 3, 4, 5, 6, 7>**

**To achieve All-Du coverage, so that each definition for each variable reaches each use for each of them.**

**- Test case: <1, 2, 3, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21>**

```
line = 1, lineNumber = 1

students = {

                name = “Alice Dodo”,

                 id = “15367901”,

                 activitiesMark = 5,

                oralPracticalMark = 5,

                midtermMark = 6,

                finalMark = 60,

                totalMark = 76,

                grade = “A+”,

                gpa = 4

                }


        Course course =

        {

        Name = “Data Base”;
        code = “12321587”;
        maxMark = 100;

        }
```

**- Test case: <1, 2, 3, 4, 5, 6, 7>**

```
line = 1, lineNumber = 0

students = {

                    name = “Alice Dodo”,

                     id = “15367901”,

                     activitiesMark = 5,

                    oralPracticalMark = 5,

                    midtermMark = 6,

                    finalMark = 60,

                    totalMark = 76,

                    grade = “A+”,

                    gpa = 4

                    }


          Course course =

          {

          Name = “Data Base”;
          code = “12321587”;
          maxMark = 100;

          }
```