```
bool filesCheck(ofstream& outFile, ifstream& inFile);
//*****************************************
// Purpose: Checks if either input or output files exists.
// Input: none.
// Pre: Declaration of both input and output streams before calling this function.
// Output: bool.
// Post: return true if files are loaded successfully, false if not.
// Note: none.
//*****************************************

void setPrecision(ofstream& outFile, int data);
//*****************************************
// Purpose: Changes quantity of numbers after decimal point.
// Input: data.
// Pre: outFile is opened & ok, data has valid value.
// Output: outFile.
// Post: Changes the quantity of numbers after decimal point to the given data in call.
// Note: none.
//*****************************************

void printHeading(ofstream& outFile);
//*****************************************
// Purpose: Print heading for output file.
// Input: none.
// Pre: outFile is opened & ok.
// Output: outFile.
// Post: Heading is printed in outFile.
// Note: none.
//*****************************************

void readRd(ifstream& inFile, string& studName, int& studId, char& sex, float& exam1, float&
exam2, float& exam3);
//*****************************************
// Purpose: Reads exams data from input file.
// Input: inFile.
// Pre: inFile is opened & ok. studName, studId, sex, exam1, exam2, exam3 have values.
// Output: studName, studId, exam1, exam2, exam3.
// Post: studName, studId, sex, exam1, exam2, exam3 are stored to their variables via inFile.
```

```
// Note: none.
//*******************************************

bool validData(int studId, char sex, float exam1, float exam2, float exam3);
//*******************************************
// Purpose: Checks if exam and student id are valid.
// Input: studId, sex, exam1, exam2, exam3.
// Pre: studId, sex, exam1, exam2, exam3 have values.
// Output: bool.
// Post: Returns True if data are valid, false is not.
// Note: none.
//*******************************************

void printRd(ofstream& outFile, string studName, int studId, char sex, float exam1, float exam2,
float exam3);
//*******************************************
// Purpose: Print intial row of student and exams data.
// Input: studName, studId, sex, exam1, exam2, exam3.
// Pre: outFile is opened & ok. studName, studId, sex, exam1, exam2, exam3 have values.
// Output: outFile.
// Post: Student name, id, sex, exam 1, exam 2, exam3 are printed in outFile.
// Note: none.
//*******************************************

float avg(float exam1, float exam2, float exam3);
//*******************************************
// Purpose: Calculates avg of 3 exams converted to the next integer.
// Input: exam1, exam2, exam3.
// Pre: exam1, exam2, exam3 are valid and have data.
// Output: float.
// Post: The avg of 3 exams converted to the next integer is returned.
// Note: none.
//*******************************************

void printAvgAndGrade(ofstream& outFile, int examAvg, char studGrade);
//*******************************************
// Purpose: Prints student avg and letter grade data.
// Input: examAvg, studGrade.
```

```
// Pre: outFile is opened & ok. examAvg, studGrade are valid and have data.
// Output: outFile.
// Post: Avg and grade letter are printed to outFile.
// Note: none.
//*****************************************

char grade(int examAvg);
//*****************************************
// Purpose: Returns the exams avg to letter grading.
// Input: examAvg.
// Pre: examAvg is valid and have data.
// Output: char.
// Post: Letter of grading is returned, grades are: A,B,C,D,F.
// Note: none.
//*****************************************

void printInvalidDataMsg(ofstream& outFile);
//*****************************************
// Purpose: Prints invalid data line.
// Input: none.
// Pre: outFile is opened & ok.
// Output: outFile.
// Post: "~~ Invalid data ~~" line is printed in outFile.
// Note: none.
//*****************************************

void countGrades(char studGrade, int& cntA, int& cntB, int& cntC, int& cntD, int& cntF);
//*****************************************
// Purpose: Depending on the input, grade count is increased by one.
// Input: studGrade.
// Pre: studGrade has value and valid grade, cnt A,B,C,D,F has values.
// Output: cntA, cntB, cntC, cntD, cntF.
// Post: Count of one of the grades cnt variables is ancreased by one, depending on the input.
// Note: none.
//*****************************************

void assignTopExam(float exam1, float exam2, float exam3, float& examTop);
//*****************************************
```

```
// Purpose: Assigning top exam value "if possible" to the examTop Variable.

// Input: exam1, exam2, exam3.

// Pre: exam1, exam2,exam3 has values, examTop is declared and has value.

// Output: examTop.

// Post: either exam1 or exam2 or exam3 is assigned to examtop if one or more of them is larger
than examTop.

// Note: none.

//*******************************************


void assignTopAvg(int studId, float examAvg, float& examAvgTop, int& examAvgTopId);

//*******************************************

// Purpose: Assigning top avg value "if possible" to the examAvgTop Variable in parallel with
examAvgTopId.

// Input: studId, examAvg.

// Pre: studId examAvg has values, examAvgTop and examAvgTopId are declared and has value.

// Output: examAvgTop, examAvgTopId.

// Post: examAvgTop is assigned as examAvg and examAvgTopId as studId Only if examAvg >
examAvgTop.

// Note: none.

//*******************************************


void maleAndFemalePrec(float studCnt, float maleCnt, float femaleCnt, float& malePerc, float&
femalePerc);

//*******************************************

// Purpose: Assigning males and females percentages according to their count.

// Input: studCnt, maleCnt, femaleCnt.

// Pre: studCnt, maleCnt, femaleCnt have values, malePerc and femalePerc are declared and has
values.

// Output: malePerc, femalePerc.

// Post: malePerc and femalePerc are assigned to their variables.

// Note: maleCnt, femaleCnt are both passed as float in order to get accurate float results in
percentages.

//*******************************************


void printBarChart(ofstream& outFile, int studCnt, int cntA, int cntB, int cntC, int cntD, int
cntF);

//*******************************************

// Purpose: Prints Grade Destitution Bar Chart.

// Input: studCnt, cntA, cntB, cntC, cntD, cntF.

// Pre: outFile is opened & ok. studCnt, cntA, cntB, cntC, cntD, cntF have values.

// Output: outFile.
```

```
// Post: grade destitution bar chart is printed to outFile.
// Note: none.
//*******************************************

void printPrec(ofstream& outFile, float malePerc, float femalePerc);
//*******************************************
// Purpose: Prints males and females percentages.
// Input: malePerc, femalePerc.
// Pre: outFile is opened & ok. malePerc, femalePerc have values.
// Output: outFile.
// Post: males and females percentages are printed to outFile.
// Note: none.
//*******************************************

void printBets(ofstream& outFile, float examTop, float examAvgTop, int examAvgTopId);
//*******************************************
// Purpose: Prints top exam performance and top avg with its student id.
// Input: examTop, examAvgTop.
// Pre: outFile is opened & ok. examTop, examAvgTop have values.
// Output: outFile.
// Post: top exam performance and top avg with its student id are printed to outFile.
// Note: none.
//*******************************************

void printNoValidDataForTopAndPerc(ofstream& outFile);
//*******************************************
// Purpose: Prints an error message if there is no valid exam records.
// Input: none.
// Pre: outFile is opened & ok.
// Output: outFile.
// Post: the meaning of "No valid student data in file to evaluate chart, percentages and top
exam and avg" is printed to outFile.
// Note: none.
//*******************************************
```