Question 1:

(3 marks) You have main function which is saved on folder named Q1 already. Use these classes and functions to test all your operations in this question, DO NOT EDIT ANY CODE in main function. You can add more operations/classes/interfaces to be able to answer the below question.

Design and code a class named **`Person`** that holds information about a **`Person`**. Information of a **`Person`** includes:

- A string describes the name of a Person.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class.

Design and code a deriving class named **`Worker`** from **`Person`** class that holds information about a **`Worker`**. Information of a **`Worker`** includes:

- A double-value holding working hours of a Worker.
- A double-value holding the payment rate for a Worker.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run and complete the below method which is declared in Worker class, the function will be used in second test case.
    - double getIncome() – use to determine salary of a person *income = salary+ income tax*, where *salary = payment rate * working hours* and
        - income tax = 10 percent out of salary if salary > 5000
        - otherwise incometax = 5 percent out of salary

The program output might look something like:

No of test case	Correct output	Correct output

2	Enter name of a worker: An An Enter payment rate of a worker: 250 Enter working hours of a worker: 150 Enter TC: 2 OUTPUT: 41250.0	Enter name of a worker: An An Enter payment rate of a worker: 80 Enter working hours of a worker: 41 Enter TC: 2 OUTPUT: 3444.0
1	Enter name of a worker: An An Enter payment rate of a worker: 250 Enter working hours of a worker: 100 Enter TC: 1 OUTPUT: An An An An 100.0 250.0	

Question 2:

(4 marks) You have main function which is saved on folder named Q2 already. Use these classes and function to test all your operations in this question, DO NOT EDIT ANY CODE in main. You can add more operations/classes/interfaces to be able to answer the below questions. We had provided you:
1. Interface – IStudent which will declare some operations for an Student – DO NOT EDIT this one.
2. **You only need to complete the code in class MyStudent and Student.**
Design and code a class named Student that holds information about an Student. Information of a Student includes:

- A string describing name of a Student.
- A string describing major of a Student

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run. Complete the method
    - public String getName() – return the name of a Student

Design and code a class named **MyStudent** which will implement interface IStudent and complete 2 methods which were declared in IStudent (`assume that number of Students in list "s" is always greater than 2).`

- List<Student> f1(List<Student> s) –. Firstly, sort the list of Student "s" ascending by student name (alphabet order), finally return the list of 2 first students in the list "s". *The sorting must ignores the case during comparison.*
- int f2(ArrayList<Student> s) – count and return the number of students in the list "s" who are in major of "SE".

The program output might look something like:

```
Enter number of Student: 3

Student name: Hoa Hoa
Student major: SE

Student name: An An
Student major: IA

Student name: Binh Binh
Student major: SE

Enter test function (1-f1;2-f2): 1

OUTPUT:
An An
Binh Binh
```

```
Enter number of Student: 3

Student name: Hoa Hoa
Student major: SE

Student name: An An
Student major: IA

Student name: Binh Binh
Student major: SE

Enter test function (1-f1;2-f2): 2

OUTPUT:
2
```

Question 3:

(3 marks) You have main function which is saved on folder named Q3 already. Use these classes and functions to test all your operations in this question, DO NOT EDIT ANY CODE in main function. You can add more operations/classes/interfaces to be able to answer the below question.

Design and code a class named **`Flight`**, that holds information of a Flight, including:

- A string describes route of a flight as the format of "*departure–destination*", for example HAN-MNL or CDG-SGN…
- A whole number describes fare of a flight.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class and complete below method which is declared in Flight class, the function will be used in second test case.
  - String getDeparture() – this function return value of departure of a route.

The program output might look something like:

No of test case	Correct output
2	Enter flight route: ICN-HAN Enter flight fare: 250 Enter TC: 2 OUTPUT: ICN
1	Enter flight route: ICN-HAN Enter flight fare: 250 Enter TC: 1 OUTPUT: ICN-HAN 250

Question 1:

(3 marks) You have main function which is saved on folder named Q1 already. Use these classes and functions to test all your operations in this question, DO NOT EDIT ANY CODE in main function. You can add more operations/classes/interfaces to be able to answer the below question.

Design and code a class named **`Person`** that holds information about a **`Person`**. Information of a **`Person`** includes:

- A string describes the name of a Person.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class.

Design and code a deriving class named **`Worker`** from **`Person`** class that holds information about a **`Worker`**. Information of a **`Worker`** includes:

- A double-value holding working hours of a Worker.
- A double-value holding the payment rate for a Worker.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run and complete the below method which is declared in Worker class, the function will be used in second test case.
    - double getIncome() – use to determine salary of a person *income = salary+ income tax*, where *salary = payment rate * working hours* and
        - income tax = 10 percent out of salary if salary > 5000
        - otherwise incometax = 5 percent out of salary

The program output might look something like:

No of test case	Correct output	Correct output

2	Enter name of a worker: An An Enter payment rate of a worker: 250 Enter working hours of a worker: 150 Enter TC: 2 OUTPUT: 41250.0	Enter name of a worker: An An Enter payment rate of a worker: 80 Enter working hours of a worker: 41 Enter TC: 2 OUTPUT: 3444.0
1	Enter name of a worker: An An Enter payment rate of a worker: 250 Enter working hours of a worker: 100 Enter TC: 1 OUTPUT: An An An An 100.0 250.0	

Question 2:

(4 marks) You have main function which is saved on folder named Q2 already. Use these classes and function to test all your operations in this question, DO NOT EDIT ANY CODE in main. You can add more operations/classes/interfaces to be able to answer the below questions. We had provided you:
1. Interface – IStudent which will declare some operations for an Student – DO NOT EDIT this one.
2. **You only need to complete the code in class MyStudent and Student.**
Design and code a class named Student that holds information about an Student. Information of a Student includes:

- A string describing name of a Student.
- A string describing major of a Student

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run. Complete the method
    - public String getName() – return the name of a Student

Design and code a class named **MyStudent** which will implement interface IStudent and complete 2 methods which were declared in IStudent (`assume that number of Students in list "s" is always greater than 2).`

- List<Student> f1(List<Student> s) –. Firstly, sort the list of Student "s" ascending by student name (alphabet order), finally return the list of 2 first students in the list "s". *The sorting must ignores the case during comparison.*
- int f2(ArrayList<Student> s) – count and return the number of students in the list "s" who are in major of "SE".

The program output might look something like:

```
Enter number of Student: 3

Student name: Hoa Hoa
Student major: SE

Student name: An An
Student major: IA

Student name: Binh Binh
Student major: SE

Enter test function (1-f1;2-f2): 1

OUTPUT:
An An
Binh Binh
```

```
Enter number of Student: 3

Student name: Hoa Hoa
Student major: SE

Student name: An An
Student major: IA

Student name: Binh Binh
Student major: SE

Enter test function (1-f1;2-f2): 2

OUTPUT:
2
```

Question 3:

(3 marks) You have main function which is saved on folder named Q3 already. Use these classes and functions to test all your operations in this question, DO NOT EDIT ANY CODE in main function. You can add more operations/classes/interfaces to be able to answer the below question.

Design and code a class named **`Flight`**, that holds information of a Flight, including:

- A string describes route of a flight as the format of "*departure–destination*", for example HAN-MNL or CDG-SGN…
- A whole number describes fare of a flight.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class and complete below method which is declared in Flight class, the function will be used in second test case.
  - String getDeparture() – this function return value of departure of a route.

The program output might look something like:

No of test case	Correct output
2	Enter flight route: ICN-HAN Enter flight fare: 250 Enter TC: 2 OUTPUT: ICN
1	Enter flight route: ICN-HAN Enter flight fare: 250 Enter TC: 1 OUTPUT: ICN-HAN 250

Question 1:

(3 marks) You have main function which is saved on folder named Q1 already. Use these classes and functions to test all your operations in this question, DO NOT EDIT ANY CODE in main function. You can add more operations/classes/interfaces to be able to answer the below question.

Design and code a class named **`Person`** that holds information about a **`Person`**. Information of a **`Person`** includes:

- A string describes the name of a Person.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class.

Design and code a deriving class named **`Worker`** from **`Person`** class that holds information about a **`Worker`**. Information of a **`Worker`** includes:

- A double-value holding working hours of a Worker.
- A double-value holding the payment rate for a Worker.

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run and complete the below method which is declared in Worker class, the function will be used in second test case.
    - double getIncome() – use to determine salary of a person *income = salary+ income tax*, where *salary = payment rate * working hours* and
        - income tax = 10 percent out of salary if salary > 5000
        - otherwise incometax = 5 percent out of salary

The program output might look something like:

No of test case	Correct output	Correct output

2	Enter name of a worker: An An Enter payment rate of a worker: 250 Enter working hours of a worker: 150 Enter TC: 2 OUTPUT: 41250.0	Enter name of a worker: An An Enter payment rate of a worker: 80 Enter working hours of a worker: 41 Enter TC: 2 OUTPUT: 3444.0
1	Enter name of a worker: An An Enter payment rate of a worker: 250 Enter working hours of a worker: 100 Enter TC: 1 OUTPUT: An An An An 100.0 250.0	

Question 2:

(4 marks) You have main function which is saved on folder named Q2 already. Use these classes and function to test all your operations in this question, DO NOT EDIT ANY CODE in main. You can add more operations/classes/interfaces to be able to answer the below questions. We had provided you:
1. Interface – IStudent which will declare some operations for an Student – DO NOT EDIT this one.
2. **You only need to complete the code in class MyStudent and Student.**
Design and code a class named Student that holds information about an Student. Information of a Student includes:

- A string describing name of a Student.
- A string describing major of a Student

Include the following member functions in your design:

- Constructors to set values for instance variables.
- Add needed operations to the class so that the main function can be run. Complete the method
    - public String getName() – return the name of a Student

Design and code a class named **MyStudent** which will implement interface IStudent and complete 2 methods which were declared in IStudent (`assume that number of Students in list "s" is always greater than 2).`

- List<Student> f1(List<Student> s) –. Firstly, sort the list of Student "s" ascending by student name (alphabet order), finally return the list of 2 first students in the list "s". *The sorting must ignores the case during comparison.*
- int f2(ArrayList<Student> s) – count and return the number of students in the list "s" who are in major of "SE".