**BBM102 – Midterm 2 (Spring 2020) – 1.6.2020   
Duration: 120 minutes**

|  |  |
| --- | --- |
| **Name Surname:** |  |
| **ID:** |  |
| **Section Number:** *C. Zalluhoğlu =1 S. Bozkir = 2 S.Sen = 3* |  |

**Questions**

**1.** Please fill in the cells according to the given statements for each row. If the statement runs correctly, it returns true or false, in this case please state the output in the fourth column. If the statement gives an error (i.e. compile or runtime error), please indicate that by typing the corresponding column (2nd or 3rd column). For each item, please write a one-sentence explanation. Note that, each row will contain only one result. (15 points in total)

**class** B { ... }

**class** C **extends** B { ... }

**class** X **extends** C { ... }

**class** Y **extends** C { ... }

X x = **new** X();

|  |  |  |  |
| --- | --- | --- | --- |
| **Statements** | **Compile Time Error** | **Runtime Error** | **Output (true/false)** |
| return (x instanceof B) ; |  |  | true |
| B b = x ; return (b instanceof C) ; |  |  | true |
| B b = x ; Y y = (Y) b ; return (y instanceof B) ; |  | class X cannot be cast to class Y |  |
| return (x instanceof Y) ; | Incompatible conditional operand types X and Y |  |  |
| Y y = x ; return (y instanceof B) ; | Type mismatch: cannot convert from X to Y |  |  |

**2.** Please write the methods f and g according to the definitions below.

The method f returns the square root of a number given as a parameter. It is known that the method f might throw exceptions in types ExceptionA and Exception B at run time, but it is not handled in the method. The method could also throw other types of exceptions at runtime, however, they are not handled in the method f either.

The method g invokes f and returns the double of its result. If the method f throws an exception in type ExceptionA, the g method returns -1. If the method f throws an exception in type ExceptionC, the g method returns 1. For any type of errors that are direct or indirect subclasses of RunTimeException, the method g returns 1. Other types of exceptions are left to the outer block for handling. (20 Points in total)

**class** ExceptionA **extends** Exception {}

**class** ExceptionB **extends** Exception {}

**class** ExceptionC **extends** RuntimeException {}

**Answer: (**Pay attention to format your code via fixed width font “Courier new” having 8pt.) **Do not forget to save the content of this box as the “code\_q2.txt” and upload to Google Forms exam form.**

|  |
| --- |
|  |

**3.** Suppose that you are managing a class having a group of students each having unique names. You want to record their scores along with their names and each student may have different number of scores. However, Map classes in Java does not permit to store duplicate keys. According to this requirement, you need to define a custom Map class named “EnhancedMap” which allows you to record different number of scores for each student. Please write the class definition and implementation of the “EnhancedMap” in concordance with the following rules: (20 points in total)

* EnhancedMap class must use the HashMap class as a backend.
* EnhancedMap class must also benefit from the generics.
* EnhancedMap must implement put, get and getMean methods
* get method must return a list of the scores for a given student name. If the given student name does not exist, you can return null value.
* getMean method returns the average of the scores for a given student name. If the given student name does not exist, you can return -1

*Quick recap for HashMap functions:*

* java.util.HashMap.put() - public V put(K key,V value) - plays role in associating the specified value with the specified key in this map. If the map previously contained a mapping for the key, the old value is replaced.
* java.util.HashMap.get() - public V get(Object key) - method returns the value to which the specified key is mapped, or null if this map contains no mapping for the key

|  |
| --- |
| **public** **static** **void** main(String []args){  System.***out***.println("Enjoying the Enhanced Map");  EnhancedMap<String,Integer> em = **new** EnhancedMap<String,Integer>();  em.put("Ahmet", 15);  em.put("Ahmet", 20);  em.put("Ahmet", 67);  System.***out***.println(em.getMean("Ahmet"));  System.***out***.println(em.get("Ahmet"));   System.***out***.println(em.getMean("Seda"));  } //Corresponding output is below  Enjoying the Enhanced Map  34.0 [15, 20, 67] -1 |

**Answer – (**Pay attention to format your code via fixed width font “Courier new” having 8pt.) **Do not forget to save the content of this box as the “code\_q3.txt” and upload to Google Forms exam form.**

|  |
| --- |
| **class** EnhancedMap<K,V> { |

**4.** In this question, you are expected to develop an “Airline Reservation System” using Java programming language. (45 points)

The requirements and rules for the system are given below:

1. “The Airline Reservation System” has three types of Person. These are Air hostess, Pilot and Passenger.
   * Each Person has name, gender and age.
   * Each Passenger has unique id and unique email
   * Each Pilot has unique license number and experience\_year.
   * Each Air Hostess has unique social security number.
2. Air hostess and Pilot are implemented by Staff. Airplane, and Bus are implemented by Vehicle.
   * Staff contains getExperience() methods.
   * Vehicle contains getModel(), getName() methods.
3. A system contains Flight.
   * Each Flight has flightNumber, departureCity, destinationCity, departureDate, destinationDate, pilot license number, hostess social security number, plane name, license number.
4. Salesman can make a reservation for a flight.
   * Salesman should enter flight number, date of flight (departureDate), seat type (vip or eco) and passengerID to make a reservation.
5. Salesman can cancel reservation for a flight.
   * Salesman should enter flight number, date of flight (departureDate) and passengerID to cancel a reservation
6. Salesman can add a passenger.
   * Salesman should enter name of Passenger, gender, email, age and passengerID to add a passenger to the system.
7. Salesman can remove a passenger.
   * Salesman should enter passengerID to remove passenger from the system. (if a passenger is assigned to a flight you cannot delete)

According to the instructions given above. You are expected:

1) A class diagram that is suitable to cover all the definitions provided above.  
2) A single .txt file including all classes and their implementations

Useful Information:

* You must use the **inheritance**, **abstract classes** and **interfaces** in order to create a design for this problem.
* You do not need to define the main object which involves the main method. However, all other classes must be sketched and typed.
* The class diagram could be drawn via draw.io or similar application and you need to save the file with .png (diagram.png) extension through setting of 100% zoom.
* The code file must be of .txt file named “code\_q4.txt”.
* It will be sufficient to define one constructor with all necessary inputs for each class.
* It will be sufficient to write down the full method signature (return value + function name + parameters) for the methods with a very brief method body involving simple println() statements. Therefore, you will not deal with business logic. Instead you will be able to focus on object design and their relations.

**Answer** – **Code** (Pay attention to format your code via fixed width font “Courier new” having 8pt.) **Do not forget to save the content of this box as the “code\_q4.txt” and upload to Google Forms exam form.**

|  |
| --- |
|  |

**Answer – Diagram** (You can add picture into this box. Please pay attention to make your drawing understandable and clearly viewable) **Do not forget to upload your diagram file (having file name diagram.jpg or diagram.png to the Google Forms exam form.**

|  |
| --- |
|  |

**Declaration of Statement** (Place the declaration of yourself into here by adding in picture format)

|  |
| --- |
|  |