Advanced Programming (CSE201), Quiz -3

Time allocated: 03:20pm – 3:50pm (30 minutes) + 5 minutes for uploading solution

Instructions:

- You must follow all the instructions sent to you earlier over the email.
- Only reasonable and clearly mentioned assumptions (if any) would be accepted.
- For justifications, please be as concise as possible (2-3 sentences only)

Question-1) Some class ABC has a method display() that could generate five different kinds of exceptions: NullPointerException, NumberFormatException, FileNotFoundException, IllegalArgumentException, and IOException. Write the try/catch block(s) inside the display() method to handle these exceptions. **[2 marks]**

Solution:

```
public void display() {
   try {
      /*statements*/
   }
   catch(NullPointerException e) { }
   catch(NumberFormatException e) { }
   /* Polymorphism -- IllegalArgumentException  is superclass of NumberFormatException.
    * Hence, NumberFormatException must be caught before IllegalArgumentException.
    * Otherwise, catch block for IllegalArgumentException will also be able to catch
    * NumberFormatException. This ordering will lead to compilation error.
    */
   catch(IllegalArgumentException e) { }
   /* The rule mentioned above holds true even here. FileNotFoundException is subclass of
    * IOException. Hence, FileNotFoundException should be caught before IOException.
    */
   catch(FileNotFoundException e) { }
   catch(IOException e) { }
}
```

Rubric:

Single try block **with** proper:

a) Catch block for NullPointerException: +0.3 marks
b) Catch block for NumberFormatException: +0.3 marks
c) Catch block for IllegalArgumentException: +0.3 marks
d) Catch block for IOException: +0.3 marks
e) Catch block for FileNotFoundException: +0.3 marks
f) FileNotFoundException caught before IOException: +0.25
g) NumberFormatException caught before IllegalArgumentException: +0.25

Question-2) Write a generic class Point that has one generic type field. Point class has a parameterized constructor to initialize this field. Your code must ensure that the generic type is not an Integer. If in case it is an Integer then InvalidTyeException should be thrown. It is a user defined exception that is extended from the RuntimeException. You should also write code for

InvalidTyeException. You should **not** show the **main** method. Write actual code. Pseudocode is not allowed. **[2 marks]**

Solution: Note main method is not to be shown.

```
public class InvalidTypeException extends RuntimeException {    //+0.25 marks
   public InvalidTypeException(String s) {      // default / parameterized constructor: +0.25 marks
      super(s);
   }
}

public class Point <T> {        //+0.25 marks
   private T field;             //+0.25 marks
   public Point(T object) {     //constructor declaration +0.25 marks
      if(object instanceof Integer) throw new InvalidTypeException("message"); //+0.75 marks
                        OR
      if(object.getClass() == Integer.class) throw new InvalidTypeException("message"); //+0.75
marks
      this.field = object;
   }
}
```

Question-3) Figure-1 shows the pseudocode for reading/writing from an IO steam. a) what is the significance of the finally block there? b) what other code can be included to allow us not to have **any** catch block here. **[3 marks]**

```
public void access_streams() {
   try {
      Open a stream
      while more information
         read (or write) information
   } finally {
      close the stream
   }
}
```

Figure-1

Solution:

Rubric: 1.5 + 1.5 marks

a) The finally block is to ensure IO steams are first closed before the control passes to the caller method when exceptions are generated inside the method access_streams.

OR

The finally block is to ensure IO steams are closed and data remaining in buffer are written into the output stream before the program terminates due to exception

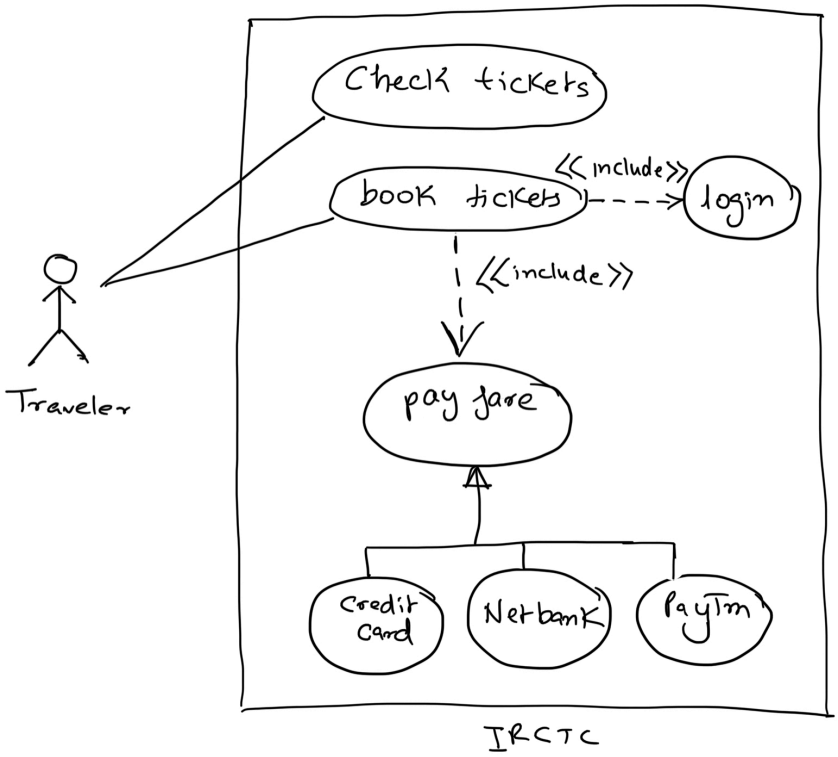
b) The access_streams method must be declared as either

```
public void access_streams() throws FileNotFoundException, IOException {
                        OR
public void access_streams() throws IOException {
                        OR
public void access_streams() throws Exception {
```

Question-4) Draw a single use case diagram for the following scenario: **[3 marks]**

"*IRCTC railway booking system allows the traveler to check the ticket availability and book the tickets. Proceeding with ticket booking would require the traveler login into IRCTC. The traveler would also be asked to pay the ticket fare while booking the ticket. Different payment mode available to the user are credit card payment, netbanking payment, and PayTM payment.*"

Solution



Rubric:
Seven use cases: 0.25 x 7 (names of use cases may vary but their intention should match)
Actor and system boundary: +0.25
Two include notation: +0.2 + 0.2
Two dashed arrows with shown arrowhead for include relationship: +0.2
Generalization relationship using proper arrows/arrowhead as shown: +0.4