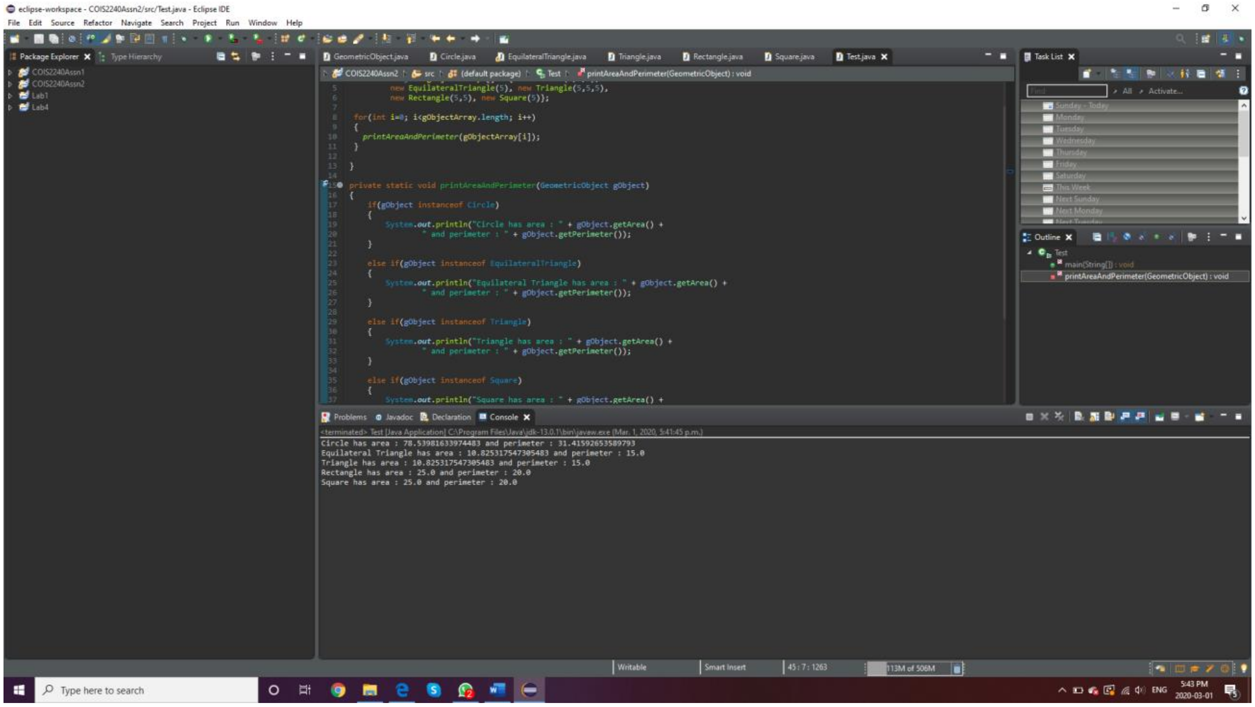
Assignment 2

Question#1: Submit the following:

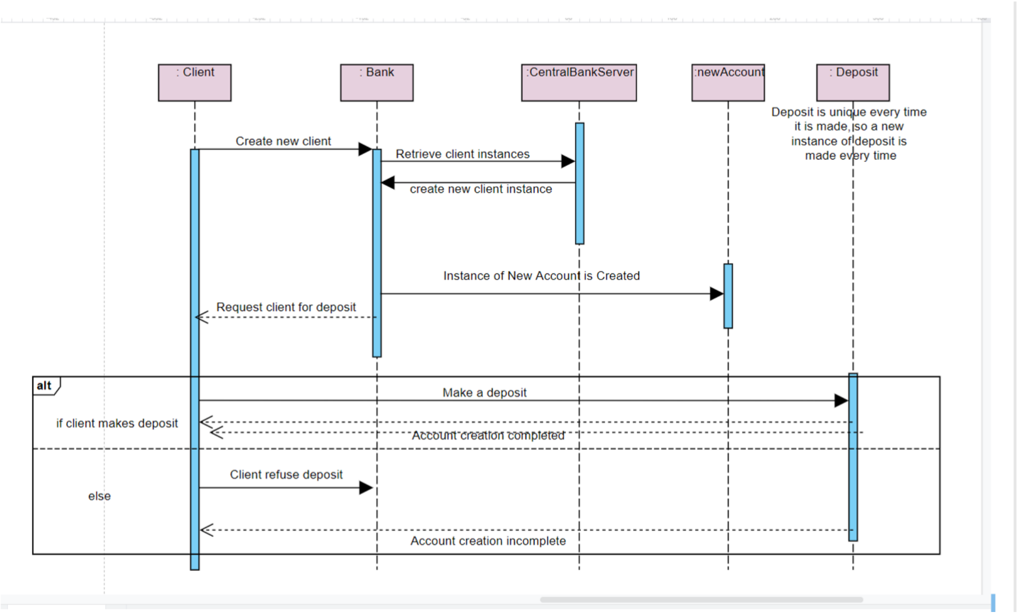
1. Implement the classes Circle, EquilateralTriangle, Rectangle, and Square.
2. Implement the method printAreaAndPerimeter in the Test class that prints the area and the perimeter of the passed GeometricObject.
3. In the Test class, create an array of GeometricObject of size 5. The first element should be assigned to a Circle object: new Circle(5,5,5). The second element should be assigned to an EquilateralTriangle object: new EquilateralTriangle(5). The third element should be assigned to a Triangle object: new Triangle(5,5,5). The fourth element should be assigned to a Rectangle object: new Rectangle(5,5). The fifth element should be assigned to a Square object: new Square(5).
4. Pass each element in the array to printAreaAndPerimeter.
5. Compile, Run, and take a screenshot of the output and submit to Blackboard (you must submit the program regardless whether it complete or incomplete, correct or incorrect)



```
            new EquilateralTriangle(5), new Triangle(5,5,5),
            new Rectangle(5,5), new Square(5)};

    for(int i=0; i<gObjectArray.length; i++)
    {
      printAreaAndPerimeter(gObjectArray[i]);
    }

  }

  private static void printAreaAndPerimeter(GeometricObject gObject)
  {
      if(gObject instanceof Circle)
      {
          System.out.println("Circle has area : " + gObject.getArea() +
                  " and perimeter : " + gObject.getPerimeter());
      }

      else if(gObject instanceof EquilateralTriangle)
      {
          System.out.println("Equilateral Triangle has area : " + gObject.getArea() +
                  " and perimeter : " + gObject.getPerimeter());
      }

      else if(gObject instanceof Triangle)
      {
          System.out.println("Triangle has area : " + gObject.getArea() +
                  " and perimeter : " + gObject.getPerimeter());
      }

      else if(gObject instanceof Square)
      {
          System.out.println("Square has area : " + gObject.getArea() +
```

```
<terminated> Test [Java Application] C:\Program Files\Java\jdk-13.0.1\bin\javaw.exe (Mar. 1, 2020, 5:41:45 p.m.)
Circle has area : 78.53981633974483 and perimeter : 31.41592653589793
Equilateral Triangle has area : 10.825317547305483 and perimeter : 15.0
Triangle has area : 10.825317547305483 and perimeter : 15.0
Rectangle has area : 25.0 and perimeter : 20.0
Square has area : 25.0 and perimeter : 20.0
```

Question#2: Draw a sequence diagram for the following scenario: a client wishes to open a new account at a bank branch. To do so, his instance of class Client must first be retrieved from the central bank server. For a new client, an instance of Client must be created. An instance of BankAccount is then created using the Client object. A deposit must then immediately follow to complete the account creation process. (5 points)

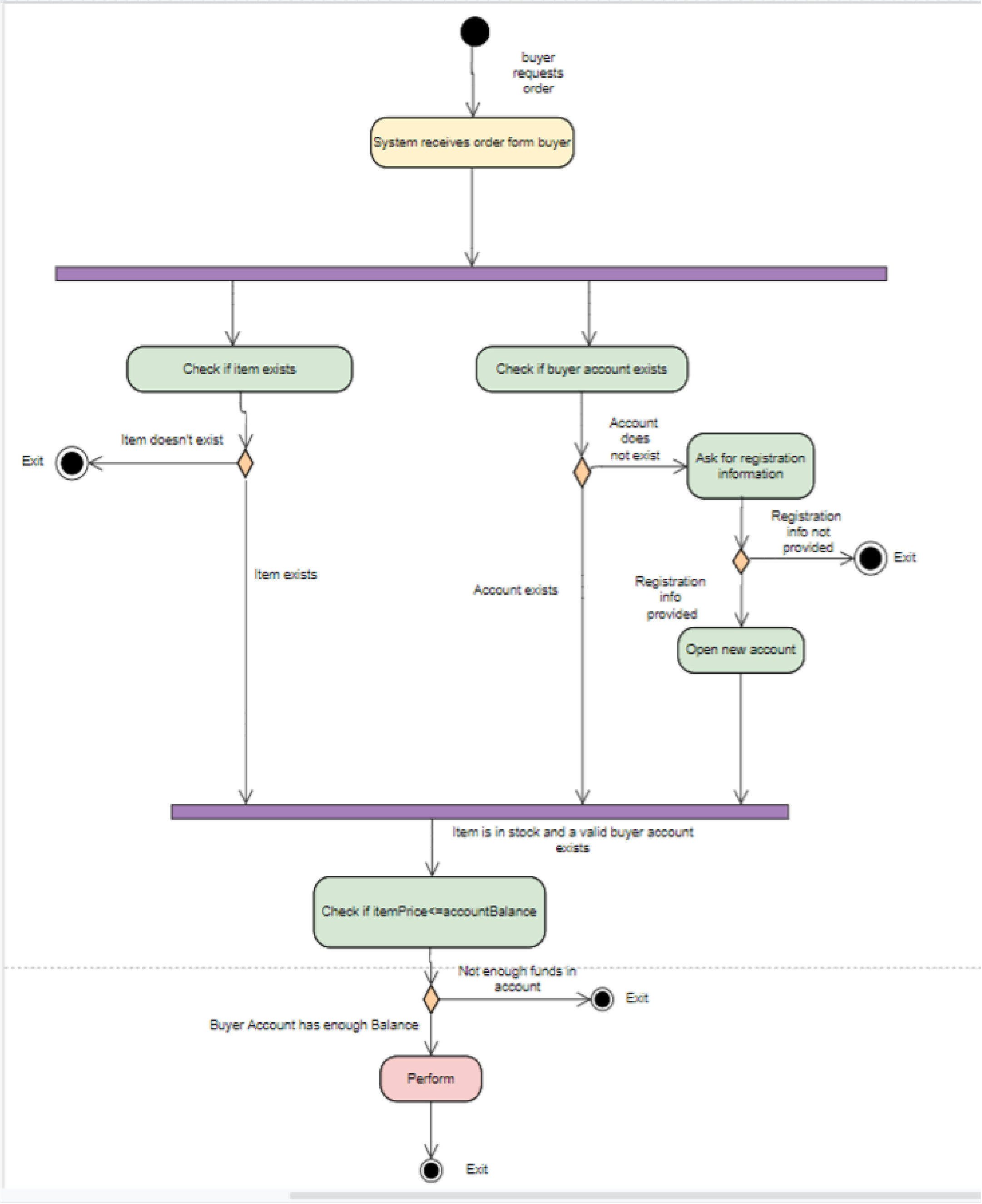


: Client
: Bank
:CentralBankServer
:newAccount
: Deposit
Deposit is unique every time it is made, so a new instance of deposit is made every time
Create new client
Retrieve client instances
create new client instance
Instance of New Account is Created
Request client for deposit
alt
Make a deposit
if client makes deposit
Account creation completed
Client refuse deposit
else
Account creation incomplete

Question#3: Draw an activity diagram for the following scenario: (5 points)

In an online purchasing system, the buyer requests to buy an item. In parallel, the system looks up whether the item exists in the store and verifies if the buyer has an account with the system. If the buyer does not have an account, the system will ask for registration info from the buyer to open an account. If the buyer does not provide registration info, the system exits. If the item does not exist in the store, the system exits. If the item exists, the system will check if the item price is less than or equal to buyer's account balance. If the buyer has enough money in the account to purchase the item, the

system completes the purchase order successfully. If the buyer does not have enough money, the system exit



buyer requests order
System receives order form buyer
Check if item exists
Check if buyer account exists
Item doesn't exist
Exit
Account does not exist
Ask for registration information
Registration info not provided
Exit
Item exists
Account exists
Registration info provided
Open new account
Item is in stock and a valid buyer account exists
Check if itemPrice<=accountBalance
Not enough funds in account
Exit
Buyer Account has enough Balance
Perform
Exit