Lab 7 Code/Output

Sol Ben-Ishay

Part I

Question 1 - College

Code:

Person.java

```
Person.java

package lab7Part1Question1;

public class Person {
   String name;
   String phoneNumber;
   String homeAddress;
   String emailAddress;

   public Person() {

   }

   public Person(String name) {
      this.name = name;
   }

   public Person (String name1, String homeAddress1, String phoneNumber1,
String emailAddress1) {
      name = name1;
      homeAddress = homeAddress1;
      phoneNumber = phoneNumber1;
      emailAddress = emailAddress1;
   }

   public void setName(String name) {
      this.name = name;
   }

   public void setHomeAddress(String homeAddress) {
      this.homeAddress = homeAddress;
   }

   public void setPhoneNumber(String phoneNumber) {
      this.phoneNumber = phoneNumber;
   }

   public void setEmailAddress(String emailAddress) {
      this.emailAddress = emailAddress;
   }




public String getName() {
      return name;
   }

   public String getHomeAddress() {
      return homeAddress;
   }

   public String getPhoneNumber() {
      return phoneNumber;
   }

   public String getEmailAddress() {
      return emailAddress;
   }

   @Override
   public String toString() {
      return "Class Name: " + this.getClass() + "Person Name: " + getName();
   }

}
```

Student.java

```
Student.java

package lab7Part1Question1;

public class Student extends Person {
   private String classStatus;

   public Student(String name) {
      super.name = name;
   }

   public Student(String name, String classStatus) {
      super.name = name;
      this.classStatus = classStatus;
   }

   public void setClassStatus(String classStatus) {
      this.classStatus = classStatus;
   }

   public String getClassStatus() {
      return classStatus;
   }
```

```
    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Person Name: " + getName();
    }

}
```

Employee.java

```
Employee.java

package lab7Part1Question1;

public class Employee extends Person {
    private String office;
    private String salary;
    private MyDate dateHired;

    public Employee(String name) {
        super.name = name;
    }

    public Employee(String name, String office, String salary, MyDate
dateHired) {
        super.name = name;
        this.office = office;
        this.salary = salary;
        this.dateHired = dateHired;
    }

    public Employee() {

    }

    public void setOffice(String office) {
        this.office = office;
    }

    public void setSalary(String salary) {
        this.salary = salary;
    }

    public void setDateHired(MyDate dateHired) {
        this.dateHired = dateHired;
    }

    public String getOffice() {
        return office;
    }

    public String getSalary() {
        return salary;
    }
    public MyDate getDateHired() {
        return dateHired;
    }

    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Person Name: " + getName();
    }

}
```

Faculty.java

```
Faculty.java

package lab7Part1Question1;

public class Faculty extends Employee {

    private String officeHours;
    private String rank;

    public Faculty(String name) {
        super();
        super.name = name;
    }

    public Faculty(String officeHours, String rank) {
        super();
        this.officeHours = officeHours;
        this.rank = rank;
    }

    public void setOfficeHours(String officeHours) {
        this.officeHours = officeHours;
    }

    public void setRank(String rank) {
        this.rank = rank;
    }

    public String getOfficeHours() {
        return officeHours;
    }

    public String getRank() {
        return rank;
    }

    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Person Name: " + getName();
    }
}
```

Staff.java

```
Staff.java

package lab7Part1Question1;

public class Staff extends Employee{
   private String title;

   public Staff(String name) {
      super.name = name;
   }

   public Staff(String name, String title) {
      this.title = title;
   }

   public void setTitle(String title) {
      this.title = title;
   }

   public String getTitle() {
      return title;
   }

   @Override
   public String toString() {
      return "Class Name: " + this.getClass() + "Person Name: " + getName();
   }

}
```

MyDate.java

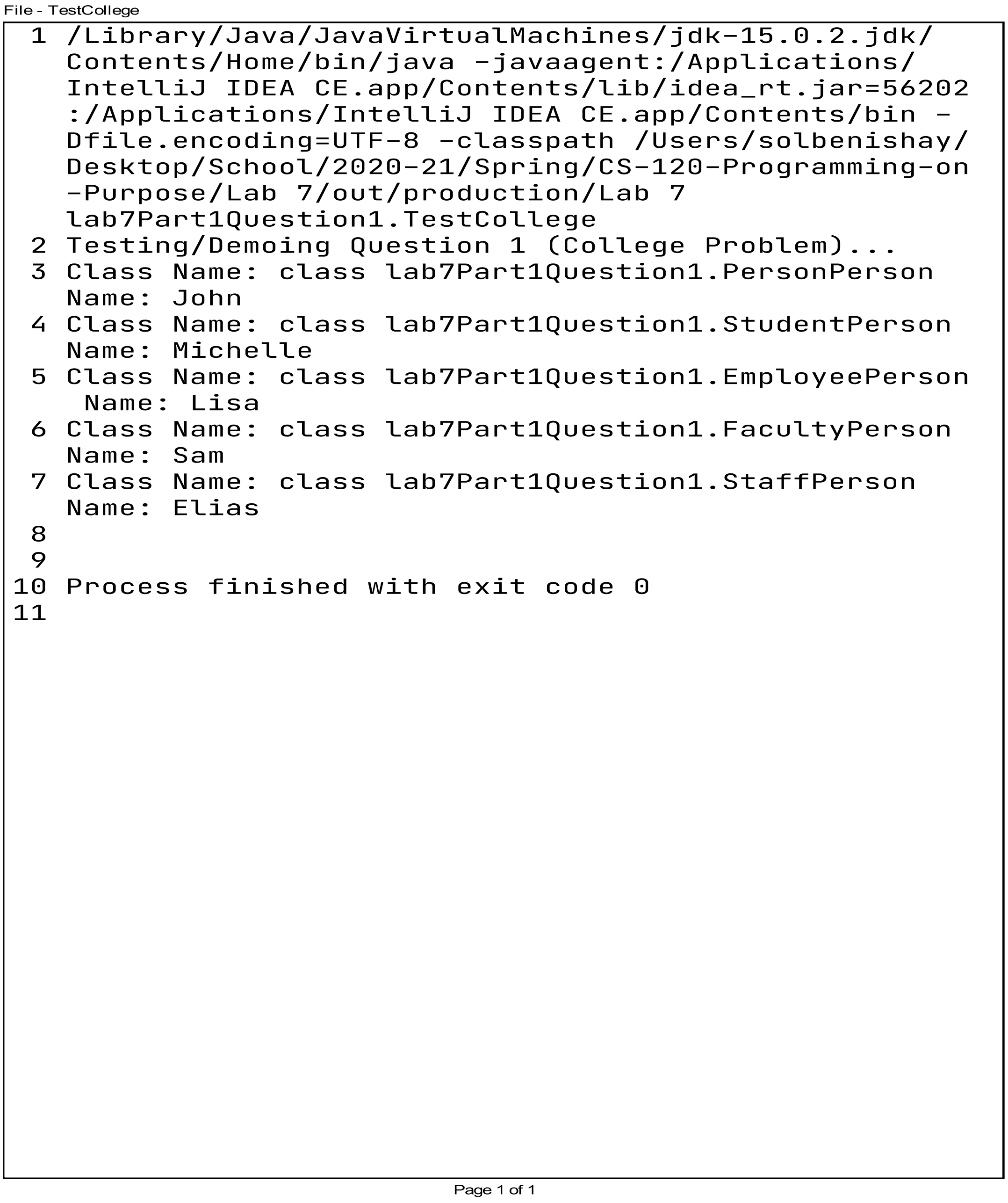
```
MyDate.java

package lab7Part1Question1;

public class MyDate {

   private int year;
   private int month;
   private int day;

   public MyDate(int dayHired, int yearHired, int monthHired) {
      day = dayHired;
      month = monthHired;
      year = yearHired;
   }
```

```
    public void setYear(int year) {
        this.year = year;
    }

    public void setMonth(int month) {
        this.month = month;
    }

    public void setDay(int day) {
        this.day = day;
    }

    public int getYear() {
        return year;
    }

    public int getMonth() {
        return month;
    }

    public int getDay() {
        return day;
    }


}
```

TestCollege.java

```
TestCollege.java

package lab7Part1Question1;

public class TestCollege {

    public static void main(String[] args) {
        System.out.println("Testing/Demoing Question 1 (College Problem)...");
        // Create College
        Person johnPerson = new Person("John");
        Student michelleStudent = new Student("Michelle");
        Employee lisaEmployee = new Employee("Lisa");
        Faculty samFaculty = new Faculty("Sam");
        Staff eliasStaff = new Staff("Elias");

        // To String Methods
        System.out.println(johnPerson.toString());
        System.out.println(michelleStudent.toString());
        System.out.println(lisaEmployee.toString());
        System.out.println(samFaculty.toString());
        System.out.println(eliasStaff.toString());
        System.out.println();
    }

}
```

Output:



File - TestCollege

```
/Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/
Contents/Home/bin/java -javaagent:/Applications/
IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=56202
:/Applications/IntelliJ IDEA CE.app/Contents/bin -
Dfile.encoding=UTF-8 -classpath /Users/solbenishay/
Desktop/School/2020-21/Spring/CS-120-Programming-on
-Purpose/Lab 7/out/production/Lab 7
lab7Part1Question1.TestCollege
Testing/Demoing Question 1 (College Problem)...
Class Name: class lab7Part1Question1.PersonPerson
Name: John
Class Name: class lab7Part1Question1.StudentPerson
Name: Michelle
Class Name: class lab7Part1Question1.EmployeePerson
 Name: Lisa
Class Name: class lab7Part1Question1.FacultyPerson
Name: Sam
Class Name: class lab7Part1Question1.StaffPerson
Name: Elias


Process finished with exit code 0

```

Page 1 of 1

Question 3 - Shape

Code:

Shape.java

```
Shape.java

package lab7Part1Question3;

public abstract class Shape {
    String shapeName;

    public Shape() {

    }

    public Shape(String name) {
        this.shapeName = name;
    }

    public void setshapeName(String name) {
        this.shapeName = name;
    }

    protected String getShapeName() {
        return shapeName;
    };

    abstract double getArea();

    abstract void drawShape();

    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Shape Name: " +
getShapeName();
    }

}
```

TwoDimShape.java

```
TwoDimShape.java

package lab7Part1Question3;

public abstract class TwoDimShape extends Shape {

    public TwoDimShape() {

    }
    @Override
    abstract double getArea();

    @Override
    abstract void drawShape();

    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Shape Name: " +
getShapeName();
    }

}
```

ThreeDimShape.java

```
ThreeDimShape.java

package lab7Part1Question3;

public abstract class ThreeDimShape extends Shape {

    public ThreeDimShape() {

    }

    @Override
    abstract double getArea();

    @Override
    abstract void drawShape();

    abstract double getVolume();

    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Shape Name: " +
getShapeName();
    }

}
```

Circle.java

```
Circle.java

package lab7Part1Question3;

public class Circle extends TwoDimShape {

    private float radius;

    // Base Constructor
    public Circle(float radius) {
        this.radius = radius;
    }

    // Constructor w/ Name
    public Circle(String name, float radius) {
        super.shapeName = name;
        this.radius = radius;
    }

    // Method to return the area
    @Override
    public double getArea() {
        return (Math.PI * Math.pow(radius, 2));
    }

    // Method to "draw" the shape
    @Override
    public void drawShape() {
        System.out.println("Drawing a circle with a radius of
"+radius+"...");
    }

    // toString method
    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Shape Name: " +
getShapeName();
    }
}
```

Recangle.java

```
Rectangle.java

package lab7Part1Question3;

public class Rectangle extends TwoDimShape {

    private float height;
    private float width;

    // Base constructor
    public Rectangle(float height, float width) {
        super();
        this.height = height;
        this.width = width;
    }




    // Constructor w/ name
    public Rectangle(String name, float height, float width) {
        super.shapeName = name;
        this.height = height;
        this.width = width;
    }

    // Method to return the area
    @Override
    public double getArea() {
        return (height * width);
    }

    // Method to "draw" the shape
    @Override
    public void drawShape() {
        System.out.println("Drawing a rectangle with a height of "+height+"
and a width of "+width+"...");
    }

    // toString method
    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Shape Name: " +
getShapeName();
    }
}
```

RectangularPrism.java

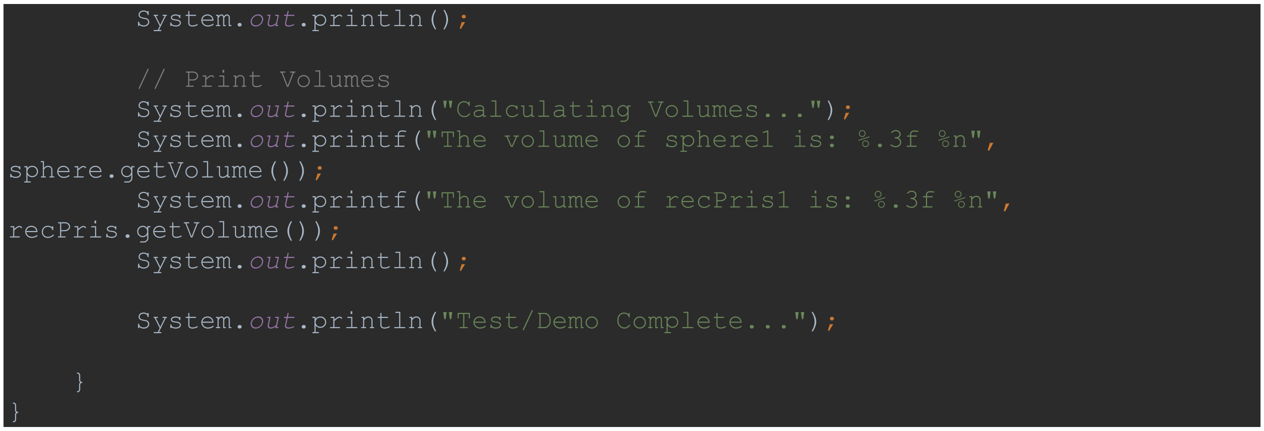
```
RectangularPrism.java

package lab7Part1Question3;

public class RectangularPrism extends ThreeDimShape {

    private float height;
    private float width;
    private float depth;

    // Base constructor
    public RectangularPrism(float height, float width, float depth) {
        this.height = height;
        this.width = width;
        this.depth = depth;
    }

    // Constructor w/ name
    public RectangularPrism(String name, float height, float width, float
depth) {
        super.shapeName = name;
        this.height = height;
        this.width = width;
        this.depth = depth;
    }
    // Method to return the area
    @Override
    public double getArea() {
        return (2*((depth*width)+(depth*height)+(width*height)));
    }

    // Method to "draw" the shape
    @Override
    public void drawShape() {
        System.out.println("Drawing a rectangular prism with a height of
"+height+", width of "+width+", and depth of "+depth+"...");
    }

    // Method to return the volume
    @Override
    public double getVolume() {
        return (height * width * depth);
    }

    // toString method
    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Shape Name: " +
getShapeName();
    }

}
```

Sphere.java

```
Sphere.java

package lab7Part1Question3;

public abstract class Shape {
    String shapeName;

    public Shape() {

    }

    public Shape(String name) {
        this.shapeName = name;
    }

    public void setshapeName(String name) {
        this.shapeName = name;
    }

    protected String getShapeName() {
        return shapeName;
    };

    abstract double getArea();
    abstract void drawShape();

    @Override
    public String toString() {
        return "Class Name: " + this.getClass() + "Shape Name: " +
getShapeName();
    }

}
```

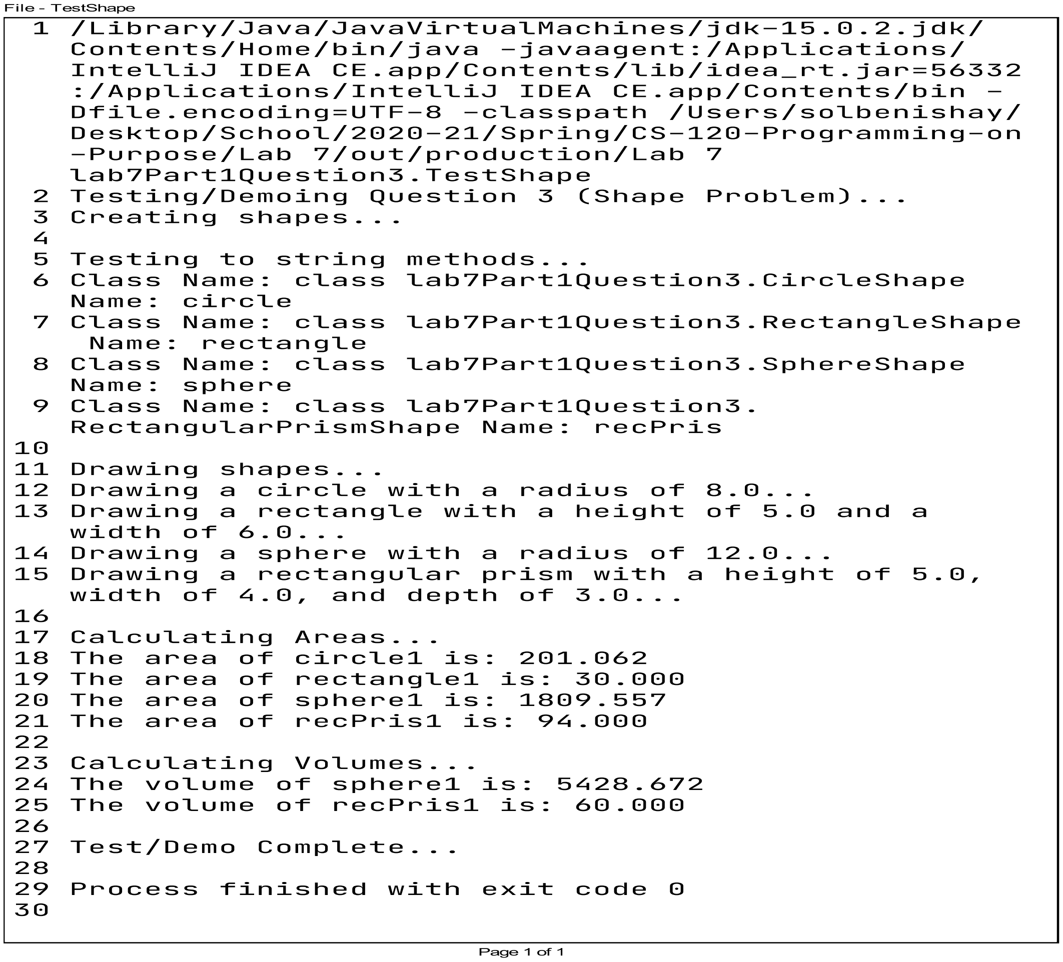
TestShape.java

```
TestShape.java

package lab7Part1Question3;

public class TestShape {
    public static void main(String[] args) {
        System.out.println("Testing/Demoing Question 3 (Shape Problem)...");
        // Create Shapes
        System.out.println("Creating shapes...");
        Circle circle = new Circle("circle",8);
        Rectangle rectangle = new Rectangle("rectangle",5,6);
        Sphere sphere = new Sphere("sphere",12);
        RectangularPrism recPris = new RectangularPrism("recPris",5,4,3);
        System.out.println();

        // To String Methods
        System.out.println("Testing to string methods...");
        System.out.println(circle.toString());
        System.out.println(rectangle.toString());
        System.out.println(sphere.toString());
        System.out.println(recPris.toString());
        System.out.println();

        // Draw Shapes (Had trouble using JavaFX Draw for 3D specifically)
        System.out.println("Drawing shapes...");
        circle.drawShape();
        rectangle.drawShape();
        sphere.drawShape();
        recPris.drawShape();
        System.out.println();

        // Print Areas
        System.out.println("Calculating Areas...");
        System.out.printf("The area of circle1 is: %.3f %n",
circle.getArea());
        System.out.printf("The area of rectangle1 is: %.3f %n",
rectangle.getArea());
        System.out.printf("The area of sphere1 is: %.3f %n",
sphere.getArea());
        System.out.printf("The area of recPris1 is: %.3f %n",
recPris.getArea());
```



```
        System.out.println();

        // Print Volumes
        System.out.println("Calculating Volumes...");
        System.out.printf("The volume of sphere1 is: %.3f %n",
sphere.getVolume());
        System.out.printf("The volume of recPris1 is: %.3f %n",
recPris.getVolume());
        System.out.println();

        System.out.println("Test/Demo Complete...");

    }
}
```

Output:



File - TestShape

```
/Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/
Contents/Home/bin/java -javaagent:/Applications/
IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=56332
:/Applications/IntelliJ IDEA CE.app/Contents/bin -
Dfile.encoding=UTF-8 -classpath /Users/solbenishay/
Desktop/School/2020-21/Spring/CS-120-Programming-on
-Purpose/Lab 7/out/production/Lab 7
lab7Part1Question3.TestShape
Testing/Demoing Question 3 (Shape Problem)...
Creating shapes...

Testing to string methods...
Class Name: class lab7Part1Question3.CircleShape
Name: circle
Class Name: class lab7Part1Question3.RectangleShape
 Name: rectangle
Class Name: class lab7Part1Question3.SphereShape
Name: sphere
Class Name: class lab7Part1Question3.
RectangularPrismShape Name: recPris

Drawing shapes...
Drawing a circle with a radius of 8.0...
Drawing a rectangle with a height of 5.0 and a
width of 6.0...
Drawing a sphere with a radius of 12.0...
Drawing a rectangular prism with a height of 5.0,
width of 4.0, and depth of 3.0...

Calculating Areas...
The area of circle1 is: 201.062
The area of rectangle1 is: 30.000
The area of sphere1 is: 1809.557
The area of recPris1 is: 94.000

Calculating Volumes...
The volume of sphere1 is: 5428.672
The volume of recPris1 is: 60.000

Test/Demo Complete...

Process finished with exit code 0

```

Page 1 of 1

Part II

Banking Problem

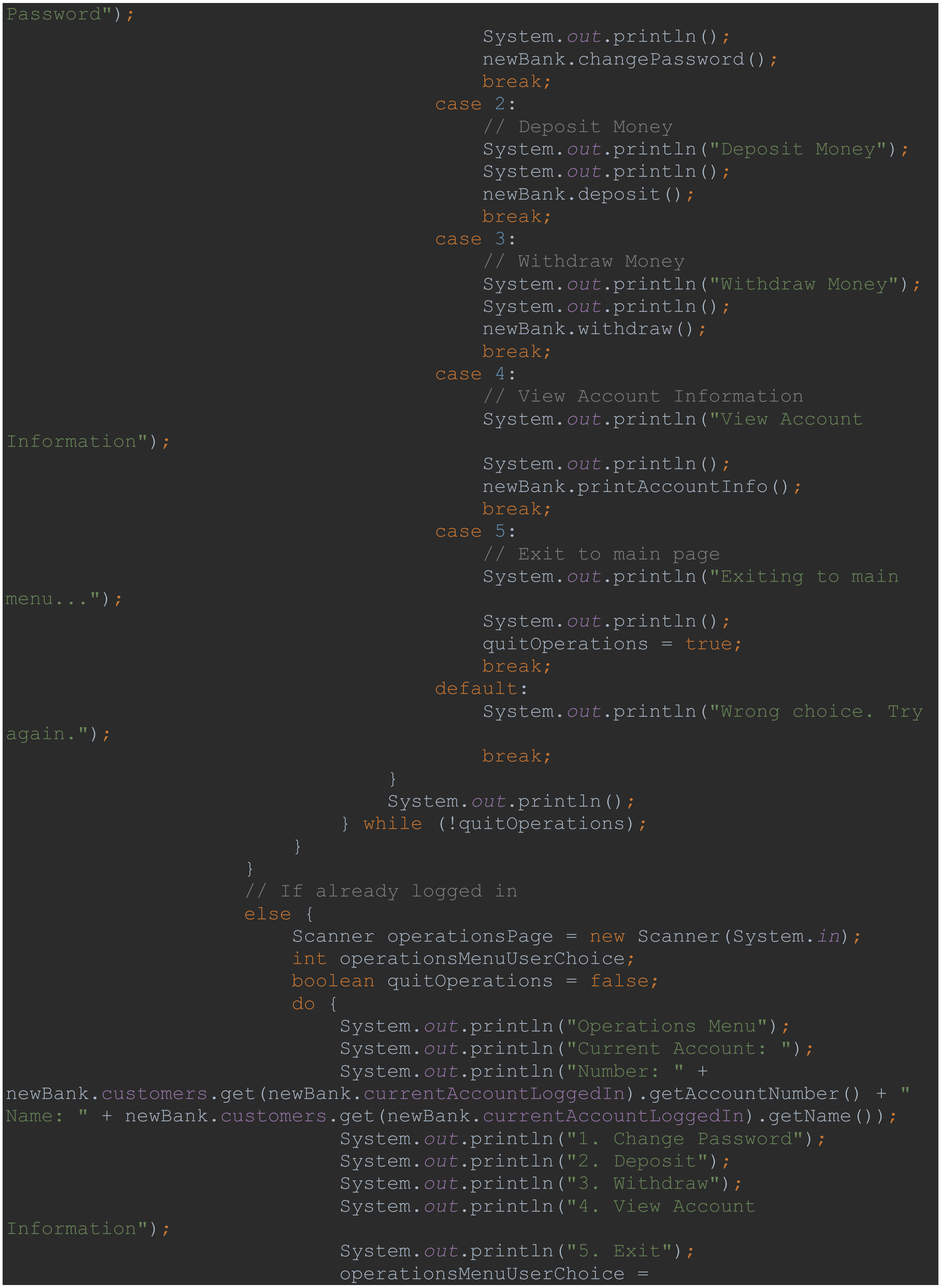
Code:

Customer.java

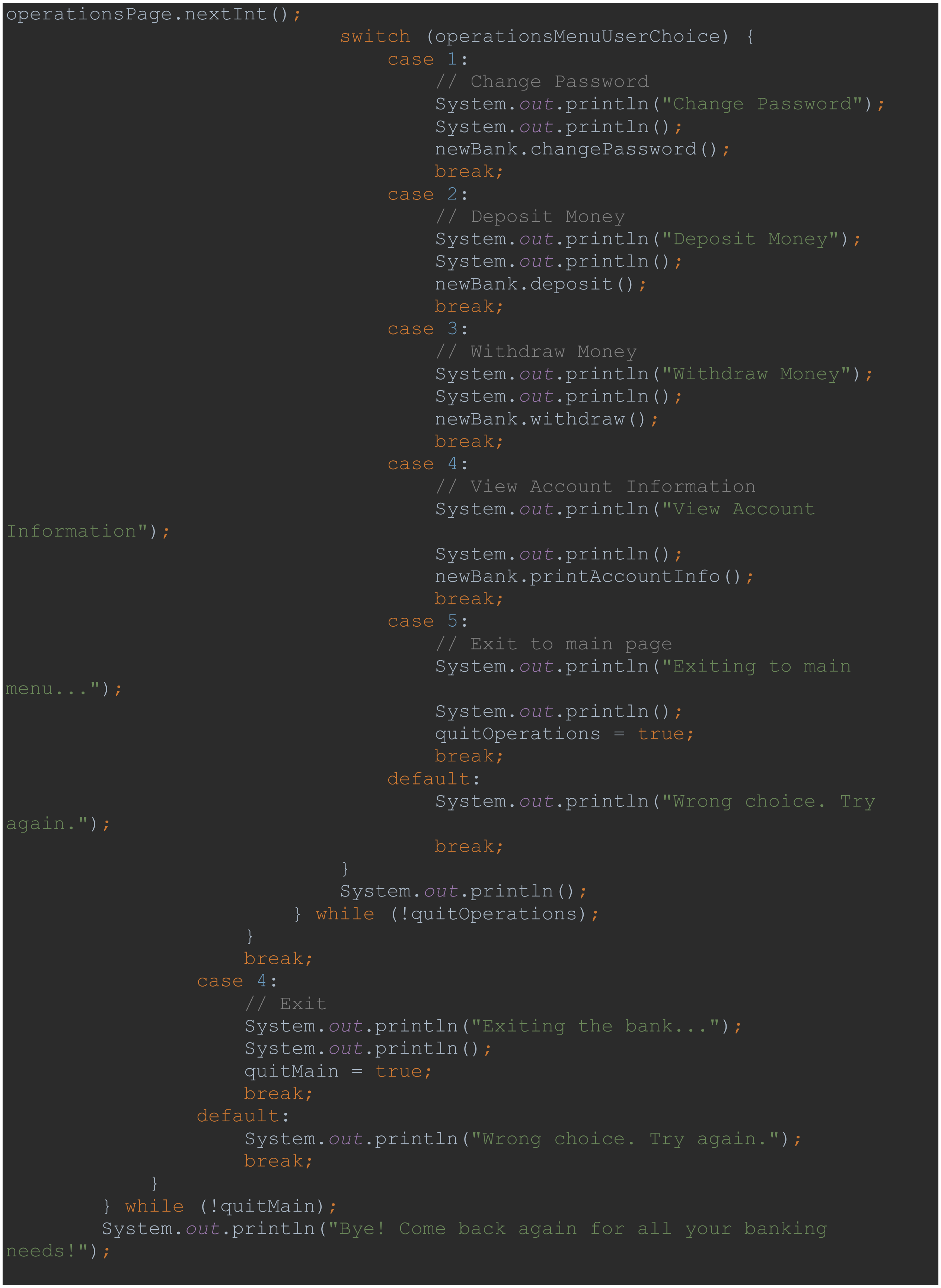
```
Customer.java

package lab7Part2;

import java.util.Scanner;

public class Customer {
    public String name;
    public String address;
    public String phoneNumber;
    public float balance;
    public int accountNumber;
    public String password;

    void setName(String name) {
        this.name = name;
    }

    void setAddress(String address) {
        this.address = address;
    }

    void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    void setBalance(float balance) {
        this.balance = balance;
    }

    void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    void setPassword(String password) {
        this.password = password;
    }

    String getName() {
        return this.name;
    }

    String getAddress() {
        return this.address;
    }

    String getPhoneNumber() {
        return this.phoneNumber;
    }

    float getBalance() {
        return this.balance;
    }

    int getAccountNumber() {
        return this.accountNumber;
    }

    String getPassword() {
        return this.password;
    }

}
```

Bank.java

```
Bank.java

package lab7Part2;

import java.util.ArrayList;
import java.util.Scanner;

public class Bank {

    // Bank variables
    int numCustomers = 0;
    int currentAccountLoggedIn = 0;
    boolean loggedIn = false;

    // ArrayList of bank customers
    ArrayList<Customer> customers = new ArrayList<Customer>();

    // Method to open a new account
    void openAccount() {
        Scanner openAccountScnr = new Scanner(System.in);
        Customer customer = new Customer();
        customer.setAccountNumber(numCustomers);
        String firstTryPass = "abc";
        String secondTryPass = "def";
        System.out.println("Create Account Number " + numCustomers + "...");
        System.out.println("Enter your name: ");
        String name = openAccountScnr.nextLine();
        customer.setName(name);
        System.out.println("Enter your address: ");
        String address = openAccountScnr.nextLine();
        customer.setAddress(address);
        System.out.println("Enter your phone number: ");
        String phoneNumber = openAccountScnr.nextLine();
        customer.setPhoneNumber(phoneNumber);
        System.out.println("How much would you like to initially deposit: ");
        float balance = openAccountScnr.nextFloat();
        customer.setBalance(balance);
        while (!firstTryPass.equals(secondTryPass)) {
            System.out.println("Enter a password: ");
            firstTryPass = openAccountScnr.next();
            System.out.println("Re-enter the password: ");
            secondTryPass = openAccountScnr.next();
            if (!firstTryPass.equals(secondTryPass)) {
                System.out.println("Passwords are not the same! Try again!");
            }
            else {
                String password = secondTryPass;
                customer.setPassword(password);
                System.out.println("Password successfully set!");
            }
        }
        customers.add(customer);
        System.out.println("Account number " + numCustomers + " successfully 
created!");
        System.out.println();
        numCustomers++;
    }

    // Method to login
    void login() {
        Scanner loginScnr = new Scanner(System.in);
        boolean accFound = false;
        int accNumber = 0;
        int loginAttempts = 3;
        while (!loggedIn && loginAttempts>0) {
            // Find Account
            while (!accFound) {
                System.out.println("Account Number: ");
                accNumber = loginScnr.nextInt();
                if (accNumber < customers.size()) {
                    accFound = true;
                }
                else if (accNumber >= customers.size()) {
                    System.out.println("Account doesn't exist. Try again!");
                    System.out.println();
                }
            }
            // Check Password
            System.out.println("Password: ");
            String password = loginScnr.next();
            if (password.equals(customers.get(accNumber).getPassword())) {
                loggedIn = true;
                currentAccountLoggedIn = accNumber;
                System.out.println("Account number " + accNumber + " 
successfully logged in!");
            }
            else if 
(!password.equals(customers.get(accNumber).getPassword())) {
                loginAttempts--;
                System.out.println("Password is not correct! " + 
loginAttempts + "attempts left!");
            }
        }
        if (loginAttempts == 0) {
            System.out.println("Too many login attempts! Returning to the
main menu!");
        }
        System.out.println();
    }

    // Method to change password
    void changePassword() {
        int attempts = 0;
        String firstTryPass = "abc";
        String secondTryPass = "def";
        boolean correctCurrentPass = false;
        Scanner custScan = new Scanner(System.in);
        while ((attempts < 3) && (correctCurrentPass == false)) {
            System.out.println("Enter your current password: ");
            String currentPassword = custScan.next();
            attempts++;
            if
(currentPassword.equals(customers.get(currentAccountLoggedIn).getPassword()))
{
                correctCurrentPass = true;
                while (!firstTryPass.equals(secondTryPass)) {
                    System.out.println("Enter your new password: ");
                    firstTryPass = custScan.next();
                    System.out.println("Re-enter your new password: ");
                    secondTryPass = custScan.next();
                    if (!firstTryPass.equals(secondTryPass)) {
                        System.out.println("Passwords are not the same! Try
again!");
                    }
                    else {

customers.get(currentAccountLoggedIn).setPassword(firstTryPass);
                        System.out.println("Password successfully set!");
                    }
                }
            }
        }
        if (attempts == 3) {
            System.out.println("Too many attempts to enter the correct
current password!");
        }
    }

    // Method to deposit money
    void deposit() {
        Scanner depScan = new Scanner(System.in);
        System.out.println("Enter the amount you would like to deposit: ");
        float depositAmount = depScan.nextFloat();
        float newBalance = customers.get(currentAccountLoggedIn).getBalance()
+ depositAmount;
        customers.get(currentAccountLoggedIn).setBalance(newBalance);
        System.out.println("Successfully deposited!");
        System.out.println("New balance: " + newBalance);
        System.out.println();
    }
    // Method to withdraw money
    void withdraw() {
        Scanner withScan = new Scanner(System.in);
        System.out.println("Enter the amount you would like to withdraw: ");
        float withdrawalAmount = withScan.nextFloat();
        float newBalance = customers.get(currentAccountLoggedIn).getBalance()
- withdrawalAmount;
        customers.get(currentAccountLoggedIn).setBalance(newBalance);
        System.out.println("Successfully withdrawn!");
        System.out.println("New balance: " + newBalance);
        System.out.println();
    }

    // Method to print account information
    void printAccountInfo() {
        System.out.println("Printing Account Information...");
        System.out.println("Account Number: " +
customers.get(currentAccountLoggedIn).getAccountNumber());
        System.out.println("Name: " +
customers.get(currentAccountLoggedIn).getName());
        System.out.println("Address: " +
customers.get(currentAccountLoggedIn).getAddress());
        System.out.println("Phone: " +
customers.get(currentAccountLoggedIn).getPhoneNumber());
        System.out.println("Balance: " +
customers.get(currentAccountLoggedIn).getBalance());
        System.out.println();
    }

    // Simulate a bank
    public static void main(String[] args) {
        System.out.println("Creating a new bank...");

        // Initialize new bank
        Bank newBank = new Bank();

        // Initialize New Customers
        Scanner bankScan = new Scanner(System.in);
        System.out.println("How many customers would you like to start with?
");
        int initNumCustomers = bankScan.nextInt();
        for (int i=0;i<initNumCustomers;i++) {
            newBank.openAccount();
        }

        System.out.println("New bank initialized with " +
newBank.numCustomers + " customers...");
        System.out.println();

        // Main Page
        Scanner mainPage = new Scanner(System.in);
        int mainMenuUserChoice;
        boolean quitMain = false;
        do {
            System.out.println("Welcome to the Bank!");
            if (newBank.loggedIn) {
                System.out.println("Current Account: ");
                System.out.println("Number: " +
newBank.customers.get(newBank.currentAccountLoggedIn).getAccountNumber() + "
Name: " + newBank.customers.get(newBank.currentAccountLoggedIn).getName());
            }
            else {
                System.out.println("Current Account: none");
            }
            System.out.println("1. New Account");
            System.out.println("2. Login");
            System.out.println("3. Operations");
            System.out.println("4. Exit");
            mainMenuUserChoice = mainPage.nextInt();
            switch (mainMenuUserChoice) {
                case 1:
                    // Create a new account
                    System.out.println("Create a new account");
                    System.out.println();
                    newBank.openAccount();
                    break;
                case 2:
                    // Login
                    System.out.println("Login");
                    System.out.println();
                    if (newBank.loggedIn) {
                        System.out.println("Logging out of account number " +
newBank.customers.get(newBank.currentAccountLoggedIn).getAccountNumber() +
"...");
                    }
                    newBank.loggedIn = false;
                    newBank.login();
                    break;
                case 3:
                    // Operations page
                    // If not already logged in
                    if (!newBank.loggedIn) {
                        System.out.println("Please login before performing
account operations...");
                        newBank.login();
                        // If logged in after attempt
                        if (newBank.loggedIn) {
                            Scanner operationsPage = new Scanner(System.in);
                            int operationsMenuUserChoice;
                            boolean quitOperations = false;
                            do {
                                System.out.println("Operations Menu");
                                System.out.println("1. Change Password");
                                System.out.println("2. Deposit");
                                System.out.println("3. Withdraw");
                                System.out.println("4. View Account
Information");
                                System.out.println("5. Exit");
                                operationsMenuUserChoice =
operationsPage.nextInt();
                                switch (operationsMenuUserChoice) {
                                    case 1:
                                        // Change Password
                                        System.out.println("Change
```



```
Password");
                                        System.out.println();
                                        newBank.changePassword();
                                        break;
                                    case 2:
                                        // Deposit Money
                                        System.out.println("Deposit Money");
                                        System.out.println();
                                        newBank.deposit();
                                        break;
                                    case 3:
                                        // Withdraw Money
                                        System.out.println("Withdraw Money");
                                        System.out.println();
                                        newBank.withdraw();
                                        break;
                                    case 4:
                                        // View Account Information
                                        System.out.println("View Account
Information");
                                        System.out.println();
                                        newBank.printAccountInfo();
                                        break;
                                    case 5:
                                        // Exit to main page
                                        System.out.println("Exiting to main
menu...");
                                        System.out.println();
                                        quitOperations = true;
                                        break;
                                    default:
                                        System.out.println("Wrong choice. Try
again.");
                                        break;
                                }
                                System.out.println();
                            } while (!quitOperations);
                        }
                    }
                    // If already logged in
                    else {
                        Scanner operationsPage = new Scanner(System.in);
                        int operationsMenuUserChoice;
                        boolean quitOperations = false;
                        do {
                            System.out.println("Operations Menu");
                            System.out.println("Current Account: ");
                            System.out.println("Number: " +
newBank.customers.get(newBank.currentAccountLoggedIn).getAccountNumber() + "
Name: " + newBank.customers.get(newBank.currentAccountLoggedIn).getName());
                            System.out.println("1. Change Password");
                            System.out.println("2. Deposit");
                            System.out.println("3. Withdraw");
                            System.out.println("4. View Account
Information");
                            System.out.println("5. Exit");
                            operationsMenuUserChoice =
```



```
operationsPage.nextInt();
                            switch (operationsMenuUserChoice) {
                                case 1:
                                    // Change Password
                                    System.out.println("Change Password");
                                    System.out.println();
                                    newBank.changePassword();
                                    break;
                                case 2:
                                    // Deposit Money
                                    System.out.println("Deposit Money");
                                    System.out.println();
                                    newBank.deposit();
                                    break;
                                case 3:
                                    // Withdraw Money
                                    System.out.println("Withdraw Money");
                                    System.out.println();
                                    newBank.withdraw();
                                    break;
                                case 4:
                                    // View Account Information
                                    System.out.println("View Account
Information");

                                    System.out.println();
                                    newBank.printAccountInfo();
                                    break;
                                case 5:
                                    // Exit to main page
                                    System.out.println("Exiting to main
menu...");

                                    System.out.println();
                                    quitOperations = true;
                                    break;
                                default:
                                    System.out.println("Wrong choice. Try
again.");

                                    break;
                            }
                            System.out.println();
                        } while (!quitOperations);
                    }
                    break;
                case 4:
                    // Exit
                    System.out.println("Exiting the bank...");
                    System.out.println();
                    quitMain = true;
                    break;
                default:
                    System.out.println("Wrong choice. Try again.");
                    break;
            }
        } while (!quitMain);
        System.out.println("Bye! Come back again for all your banking
needs!");
```

```
        }

}
```

Output:

```
/Library/Java/JavaVirtualMachines/jdk-15.0.2.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ IDEA CE.app/Contents/lib/idea_rt.jar=56414:/Applications/IntelliJ IDEA CE.app/Contents/bin -Dfile.encoding=UTF-8 -classpath /Users/solbenishay/Desktop/School/2020-21/Spring/CS-120-Programming-on-Purpose/Lab 7/out/production/Lab 7 lab7Part2.Bank
Creating a new bank...
How many customers would you like to start with?
1
Create Account Number 0...
Enter your name:
Sol Ben-Ishay
Enter your address:
540 Homans Avenue
Enter your phone number:
201-214-7024
How much would you like to initially deposit:
100
Enter a password:
sol1818
Re-enter the password:
sol1818
Password successfully set!
Account number 0 successfully created!

New bank initialized with 1 customers...

Welcome to the Bank!
Current Account: none
1. New Account
2. Login
3. Operations
4. Exit
2
Login

Account Number:
0
Password:
sol1818
Account number 0 successfully logged in!

Welcome to the Bank!
```

```
Current Account:
Number: 0 Name: Sol Ben-Ishay
1. New Account
2. Login
3. Operations
4. Exit
3
Operations Menu
Current Account:
Number: 0 Name: Sol Ben-Ishay
1. Change Password
2. Deposit
3. Withdraw
4. View Account Information
5. Exit
2
Deposit Money

Enter the amount you would like to deposit:
20.20
Successfully deposited!
New balance: 120.2


Operations Menu
Current Account:
Number: 0 Name: Sol Ben-Ishay
1. Change Password
2. Deposit
3. Withdraw
4. View Account Information
5. Exit
4
View Account Information

Printing Account Information...
Account Number: 0
Name: Sol Ben-Ishay
Address: 540 Homans Avenue
Phone: 201-214-7024
Balance: 120.2


Operations Menu
```

```
Current Account:
Number: 0 Name: Sol Ben-Ishay
1. Change Password
2. Deposit
3. Withdraw
4. View Account Information
5. Exit
5
Exiting to main menu...


Welcome to the Bank!
Current Account:
Number: 0 Name: Sol Ben-Ishay
1. New Account
2. Login
3. Operations
4. Exit
4
Exiting the bank...

Bye! Come back again for all your banking needs!

Process finished with exit code 0

```