

Problem 1

```
package Day3.lab1;

public class Address {
    private String street;
    private String city;
    private String state;
    private String zip;

    public Address(String street, String city, String state, String zip) {
        this.street = street;
        this.city = city;
        this.state = state;
        this.zip = zip;
    }

    public String getStreet() {
        return street;
    }

    public void setStreet(String street) {
        this.street = street;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public String getState() {
        return state;
    }

    public void setState(String state) {
        this.state = state;
    }

    public String getZip() {
        return zip;
    }

    public void setZip(String zip) {
        this.zip = zip;
    }
}
```

```
package Day3.lab1;

public class Customer {
    private String fName;
    private String lName;
    private String ssn;
    private Address billingAddress;
    private Address shippingAddress;

    public Customer(String fName, String lName, String ssn) {
        this.fName = fName;
        this.lName = lName;
        this.ssn = ssn;
    }

    public String getfName() {
        return fName;
    }

    public void setfName(String fName) {
        this.fName = fName;
    }

    public String getlName() {
        return lName;
    }

    public void setlName(String lName) {
        this.lName = lName;
    }

    public String getSsn() {
        return ssn;
    }

    public void setSsn(String ssn) {
        this.ssn = ssn;
    }

    public Address getBillingAddress() {
        return billingAddress;
    }

    public void setBillingAddress(Address billingAddress) {
        this.billingAddress = billingAddress;
    }

    public Address getShippingAddress() {
        return shippingAddress;
    }

    public void setShippingAddress(Address shippingAddress) {
        this.shippingAddress = shippingAddress;
    }
}
```

```

    public String toString() {
        return "[" + fName + ", " + lName + ", " + "ssn: " + ssn + "]";
    }
}

package Day3.lab1;

public class MainTest {

    public static void main(String[] args) {

        Customer c1 = new Customer("Aya", "Mohamed", "123");
        Customer c2 = new Customer("yussuf", "Mohamed", "456");
        Customer c3 = new Customer("yehia", "Mohamed", "789");

        Address address1 = new Address("street1", "city1", "chicago", "52557");
        Address address2 = new Address("street2", "city2", "minnisota",
"52557");
        Address address3 = new Address("street3", "city3", "iowa", "52557");

        c1.setBillingAddress(address1);
        c1.setShippingAddress(address1);

        c2.setBillingAddress(address2);
        c2.setShippingAddress(address2);

        c3.setBillingAddress(address3);
        c3.setShippingAddress(address3);

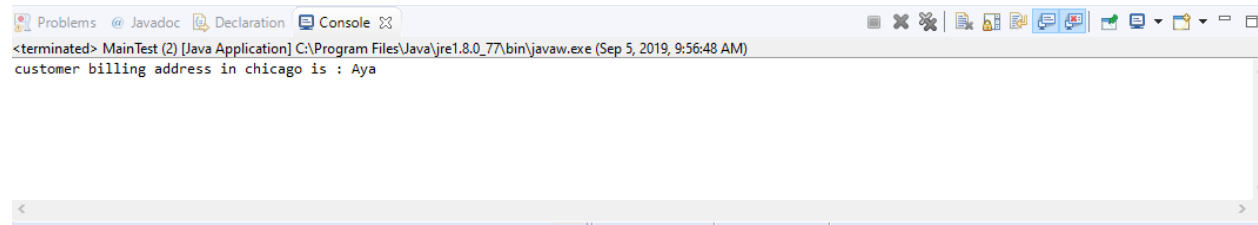
        Customer[] customers = { c1, c2, c3 };
        for (int i = 0; i < customers.length; i++) {
            if
(customers[i].getBillingAddress().getState().equals("chicago")) {
                System.out.println("customer billing address in chicago is
: " + customers[i].getfName());
            }
        }

    }

}

```

Output



The screenshot shows a Java IDE window with a console output. The title bar includes 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console text reads: '<terminated> MainTest (2) [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\javaw.exe (Sep 5, 2019, 9:56:48 AM)' followed by 'customer billing address in chicago is : Aya'.

```

<terminated> MainTest (2) [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\javaw.exe (Sep 5, 2019, 9:56:48 AM)
customer billing address in chicago is : Aya

```

Problem 4

```
package Day3.lab2;
```

```
final public class Circle {  
  
    private final double r;  
    private final double pi = 3.14;  
  
    public Circle(double r) {  
        this.r = r;  
    }  
  
    public double getR() {  
        return r;  
    }  
  
    public double getPi() {  
        return pi;  
    }  
  
    public double computeArea() {  
        return pi * r * r;  
    }  
}
```

```
package Day3.lab2;
```

```
final public class Rectangle {  
  
    private final double width;  
    private final double height;  
  
    public Rectangle(double width, double height) {  
        this.width = width;  
        this.height = height;  
    }  
  
    public double computeArea() {  
        return width * height;  
    }  
  
    public double getWidth() {  
        return width;  
    }  
  
    public double getHeight() {  
        return height;  
    }  
}
```

```

package Day3.lab2;

final public class Triangle {

    private final double height, base;

    public Triangle(double height, double base) {
        this.height = height;
        this.base = base;
    }

    public double getHeight() {
        return height;
    }

    public double getBase() {
        return base;
    }

    public double computeArea() {
        return 0.5 * height * base;
    }
}

```

```

package Day3.lab2;

import java.util.Scanner;

public class MainTest {

    public static void main(String[] args) {

        System.out.println("Enter C for Circle Enter R for Rectangle Enter T for
Triangle");

        Scanner scanner = new Scanner(System.in); // Create a Scanner object

        String choice = scanner.nextLine();

        switch (choice) {

            case "T":
                System.out.println("*****Triangle*****");

                System.out.println("please enter height");

                double height = scanner.nextDouble();

                System.out.println("please enter base");

                double base = scanner.nextDouble();

                Triangle tri = new Triangle(height, base);

```

```

        System.out.println("triangle area= " + tri.computeArea());

        System.out.println();

        break;
    case "R":
        System.out.println("*****Rectangle*****");

        System.out.println("please enter width");

        double width = scanner.nextDouble();

        System.out.println("please enter height");

        double recHight = scanner.nextDouble();

        Rectangle rec = new Rectangle(width, recHight);

        System.out.println("rectangle area= " + rec.computeArea());

        break;
    case "C":
        System.out.println("*****Circle*****");

        System.out.println("please enter radius");

        double radius = scanner.nextDouble();

        Circle cir = new Circle(radius);

        System.out.println("circle area= " + cir.computeArea());

        System.out.println();

        break;
    default:
        System.out.println("Invalid response");
    }

    scanner.close();

}

}

```

Output

<terminated> MainTest (3) [Java Application] C:\Program Files\Java\jre1.8.0_77\bin\javaw.exe (Sep 5, 2019, 10:50:45 AM)

Enter C for Circle Enter R for Rectangle Enter T for Triangle

C

*****Circle*****

please enter radius

20

circle area= 1256.0

Enter C for Circle Enter R for Rectangle Enter T for Triangle

T

*****Triangle*****

please enter height

10

please enter base

20

triangle area= 100.0

Enter C for Circle Enter R for Rectangle Enter T for Triangle

R

*****Rectangle*****

please enter width

10

please enter height

20

rectangle area= 200.0

Problem 2

```
package day4problem1;
```

```
public class Account {
    private final static double DEFAULT_BALANCE = 0.0;
    private double balance;
    private AccountType acctType;
    private Employee employee;

    Account(Employee emp, AccountType acctType, double balance) {
        employee = emp;
        this.acctType = acctType;
        this.balance = balance;
    }

    Account(Employee emp, AccountType acctType) {
        this(emp, acctType, DEFAULT_BALANCE);
    }

    public String toString() {
```

```

        return "type = " + acctType + ", balance = " + balance;
    }

    public void makeDeposit(double deposit) {
        balance += deposit;
    }

    public boolean makeWithdrawal(double amount) {
        if (amount > balance)
            return false;
        else
            balance -= amount;

        return true;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public AccountType getAcctType() {
        return acctType;
    }

    public void setAcctType(AccountType acctType) {
        this.acctType = acctType;
    }
}

package day4problem1;

public enum AccountType {
    CHECKING, SAVINGS, RETIREMENT;
}

```



```
package day4problem1;
```

```
import java.util.Date;
```

```
import java.util.GregorianCalendar;
```

```
public class Employee {
```

```
    // instance fields
```

```
    private String name;
```

```
    private String nickName;
```

```
    private double salary;
```

```
    private Date hireDay;
```

```
    // constructor
```

```
    Employee(String name, String aNickName, double aSalary, int aYear, int aMonth, int aDay) {
```

```
        this.name = name;
```

```
        nickName = aNickName;
```

```
        salary = aSalary;
```

```
        GregorianCalendar cal = new GregorianCalendar(aYear, aMonth - 1, aDay);
```

```
        hireDay = cal.getTime();
```

```
    }
```

```
    // instance methods
```

```
    public String getName() {
```

```
        return name;
```

```
    }
```

```
    public String getNickName() {
```

```
        return nickName;
```

```
    }
```

```
    public void setNickName(String aNickName) {
```

```

        nickName = aNickName;
    }

    public double getSalary() {
        return salary;
    }

    // needs to be improved
    public Date getHireDay() {
        return (Date) hireDay.clone();
    }

    public void raiseSalary(double byPercent) {
        double raise = salary * byPercent / 100;
        salary += raise;
    }

    private String format = "name = %s, salary = %.2f, hireDay = %s";

    public String toString() {
        return String.format(format, name, salary, Util.dateAsString(hireDay));
    }
}

```

```

package day4problem1;

public class MainTest {

    public static void main(String[] args) {

        Employee emp = new Employee("Aya", "yussuf", 1000, 2019, 03, 17);

        Account acc = new Account(emp, AccountType.CHECKING, 300);

        Account acc2 = new Account(emp, AccountType.SAVINGS, 300);

        Account acc3 = new Account(emp, AccountType.RETIEMENT, 300);

        acc.makeDeposit(5000); // in checking account
        acc2.makeDeposit(4000); // in saving account
        acc3.makeDeposit(3000); // in retirement account

        System.out.println(acc.toString());
        System.out.println(acc2.toString());
        System.out.println(acc3.toString());

        System.out.println(acc.makeWithdrawal(6000)); // false because the amount
we want to withdraw is bigger than you have
        System.out.println(acc.makeWithdrawal(2000)); // true because it has
enough money
        System.out.println(acc.makeWithdrawal(500)); // true because it has
enough money
    }

}

```

```

type = CHECKING, balance = 5300.0
type = SAVINGS, balance = 4300.0
type = RETIREMENT, balance = 3300.0
false
true
true

```

Problem 3

```

package day4problem3.employeeinfo;

public class Account {
    private final static double DEFAULT_BALANCE = 0.0;
    private double balance;
    private AccountType acctType;
    private Employee employee;

    Account(Employee emp, AccountType acctType, double balance) {
        this.employee = emp;
    }
}

```

```

        this.acctType = acctType;
        this.balance = balance;
    }

    Account(Employee emp, AccountType acctType) {
        this(emp, acctType, DEFAULT_BALANCE);
    }

    public String toString() {
        return "type = " + acctType + ", balance = " + balance;
    }

    public void makeDeposit(double deposit) {
        balance += deposit;
    }

    public boolean makeWithdrawal(double amount) {
        if (amount > balance)
            return false;
        else
            balance -= amount;
        return true;
    }

    public double getBalance() {
        return balance;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public AccountType getAcctType() {
        return acctType;
    }

    public void setAcctType(AccountType acctType) {
        this.acctType = acctType;
    }
}

package day4problem3.employeeinfo;

public enum AccountType {
    CHECKING, SAVINGS, RETIREMENT;
}

package day4problem3.employeeinfo;

import java.time.LocalDate;

public class Employee {

```

```

private Account savingsAcct;
private Account checkingAcct;
private Account retirementAcct;
private String name;
private LocalDate hireDate;

public Employee(String name, int yearOfHire, int monthOfHire, int dayOfHire) {
    this.name = name;
    /*
     * update, using LocalDate GregorianCalendar cal = new
     * GregorianCalendar(yearOfHire,monthOfHire-1,dayOfHire); hireDate =
     * cal.getTime();
     */
    this.hireDate = hireDate.of(yearOfHire, monthOfHire, monthOfHire);
}

public void createNewChecking(double startAmount) {
    // implement
    checkingAcct = new Account(this, AccountType.CHECKING, startAmount);
}

public void createNewSavings(double startAmount) {
    // implement
    savingsAcct = new Account(this, AccountType.SAVINGS, startAmount);
}

public void createNewRetirement(double startAmount) {
    // implement
    retirementAcct = new Account(this, AccountType.RETIREMENT, startAmount);
}

public String getFormattedAcctInfo() {
    // implement
    String message = "";
    if (checkingAcct != null) {
        message += checkingAcct.toString() + "\t";
    }
    if (savingsAcct != null) {
        message += savingsAcct.toString() + "\t";
    }
    if (retirementAcct != null) {
        message += retirementAcct.toString() + "\t";
    }

    return message;
}

public void deposit(AccountType acctType, double amt) {
    // implement
    switch (acctType) {
        case CHECKING:
            checkingAcct.makeDeposit(amt);
    }
}

```

```

        break;
    case SAVINGS:
        savingsAcct.makeDeposit(amt);
        break;
    case RETIREMENT:
        retirementAcct.makeDeposit(amt);
        break;
    default:
        break;
    }
}

public boolean withdraw(AccountType acctType, double amt) {
    // implement
    switch (acctType) {
        case CHECKING:
            return checkingAcct.makeWithdrawal(amt);
        case SAVINGS:
            return savingsAcct.makeWithdrawal(amt);
        case RETIREMENT:
            return retirementAcct.makeWithdrawal(amt);
        default:
            break;
    }
    return false;
}

public String getName() {
    return name;
}

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

public LocalDate getHireDate() {
    return hireDate;
}

public void setHireDate(LocalDate hireDate) {
    this.hireDate = hireDate;
}
}

```

```

package day4problem3.employeeinfo;

import java.util.Scanner;

public class Main {
    Employee[] emps = null;

    public static void main(String[] args) {
        new Main();
    }

    Main() {
        emps = new Employee[3];
        emps[0] = new Employee("Jim Daley", 2000, 9, 4);
        emps[1] = new Employee("Bob Reuben", 1998, 1, 5);
        emps[2] = new Employee("Susan Randolph", 1997, 2, 13);

        emps[0].createNewChecking(10500);
        emps[0].createNewSavings(1000);
        emps[0].createNewRetirement(9300);
        emps[1].createNewChecking(34000);
        emps[1].createNewSavings(27000);
        emps[2].createNewChecking(10038);
        emps[2].createNewSavings(12600);
        emps[2].createNewRetirement(9000);

        Scanner sc = new Scanner(System.in);
        System.out.print("See a report of all account balances? (y/n) ");
        String answer = sc.next();
        if (answer.equalsIgnoreCase("y")) {
            String info = getFormattedAccountInfo();
            System.out.println(info);
        } else {
            // do nothing..the application ends here
            System.exit(0);
        }
    }

    String getFormattedAccountInfo() {
        // loop through employees array and get formatted
        // account info for each employee, and assemble into a string
        String message = "";
        for (Employee emp : emps) {
            message += "\n\n" + emp.getName() + "\n";
            message += emp.getFormattedAcctInfo();
        }
        return message;
    }
}

```

Output

See a report of all account balances? (y/n) y

Jim Daley

type = CHECKING, balance = 10500.0 type = SAVINGS, balance = 1000.0 type = RETIREMENT, balance = 9300.0

Bob Reuben

type = CHECKING, balance = 34000.0 type = SAVINGS, balance = 27000.0

Susan Randolph

type = CHECKING, balance = 10038.0 type = SAVINGS, balance = 12600.0 type = RETIREMENT, balance = 9000.0