

# OBJECT ORIENTED PROGRAMMING IN JAVA - EXERCISES

## CHAPTER 1

### 1. Write Text-Based Application using Object-Oriented Approach to display your name.

```
// filename: Name.java
// Class containing display() method, notice the class doesn't have a main() method
public class Name {
    public void display() {
        System.out.println("Mohamed Faisal");
    }
}

// filename: DisplayName.java
// place in same folder as the Name.java file
// Class containing the main() method
public class DisplayName {
    public static void main(String[] args) {
        Name myname = new Name(); // creating a new object of Name class
        myname.display(); // executing the display() method in the Name class
    }
}
```

### 2. Write a java Applet to display your age.

```
// filename: DisplayNameApplet.java
import java.applet.Applet; // import necessary libraries for an applet
import java.awt.Graphics;
public class DisplayNameApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("Mohamed Faisal", 50, 25);
    }
}

// filename: DisplayNameApplet.htm
// place in same folder as the compiled DisplayNameApplet.class file
<HTML>
    <HEAD>
        <TITLE> Displaying my Name </TITLE>
    </HEAD>
    <BODY>
        <APPLET CODE="DisplayNameApplet.class" WIDTH=150 HEIGHT=25></APPLET>
    </BODY>
</HTML>
```

## CHAPTER 2

### 3. Write a program that calculates and prints the product of three integers.

```
// filename: Q1.java
import java.util.Scanner; // import Scanner libraries for input

public class Q1 {

    public static void main(String[] args) {
        Scanner input = new Scanner (System.in);

        int number1;
        int number2;
        int number3;

        System.out.println("Enter the First Number");
```

```

        number1 = input.nextInt();

        System.out.println("Enter the Second Number");
        number2 = input.nextInt();

        System.out.println("Enter the Third Number");
        number3 = input.nextInt();

        System.out.printf("The product of three number is %d:", number1 * number2 *
number3);
    }

}

```

**4. Write a program that converts a Fahrenheit degree to Celsius using the formula:**

$$celsius = \left(\frac{5}{9}\right)(fahrenheit - 32)$$

```

// filename: Q2.java
import java.util.*;

public class Q2 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        double celsius;
        double tempInFahrenheit = 0.0;

        celsius = (tempInFahrenheit - 32.0) * 5.0 / 9.0;
        System.out.println("Enter the fahrenheit value");
        tempInFahrenheit = input.nextDouble();

        System.out.printf("The celsius value of %10.2f is %2.2f",tempInFahrenheit,
celsius);
    }

}

```

**5. Write an application that displays the numbers 1 to 4 on the same line, with each pair of adjacent numbers separated by one space. Write the application using the following techniques:**

- a. Use one System.out.println statement.
- b. Use four System.out.print statements.
- c. Use one System.out.printf statement.

```

// filename: Printing.java
public class Printing {

    public static void main(String[] args) {
        int num1 = 1;
        int num2 = 2;
        int num3 = 3;
        int num4 = 4;

        System.out.println(num1 + " " + num2 + " " + num3 + " " + num4);

        System.out.print(num1 + " " + num2 + " " + num3 + " " + num4);

        System.out.printf("\n%d %d %d %d",num1,num2,num3,num4);
    }

}

```

6. Write an application that asks the user to enter two integers, obtains them from the user and prints their sum, product, difference and quotient (division).

```
// File: NumberCalc1.java
import java.util.Scanner; // include scanner utility for accepting keyboard input

public class NumberCalc1 { // begin class

    public static void main(String[] args) { // begin the main method
        Scanner input=new Scanner (System.in); //create a new Scanner object to use
        int num1=0, num2=0; // initialize variables
        System.out.printf("NUMBER CALCULATIONS\n\n");

        System.out.printf("Enter First Number:\t ");
        num1=input.nextInt(); // store next integer in num1

        System.out.printf("Enter Second Number:\t ");
        num2=input.nextInt(); // store next integer in num2

        // display the sum, product, difference and quotient of the two numbers
        System.out.printf("-----\n");
        System.out.printf("\tSum =\t\t %d\n", num1+num2);
        System.out.printf("\tProduct =\t %d\n", num1*num2);
        System.out.printf("\tDifference =\t %d\n", num1-num2);
        System.out.printf("\tQuotient =\t %d\n", num1/num2);
    }
}
```

## CHAPTER 3

7. Write an application that asks the user to enter two integers, obtains them from the user and displays the larger number followed by the words “is larger”. If the numbers are equal, print “These numbers are equal”

```
// File: Question1.java
// Author: Abdulla Faris

import java.util.Scanner; // include scanner utility for accepting keyboard input

public class Question1 { // begin class

    public static void main(String[] args) { // begin the main method
        Scanner input=new Scanner (System.in); //create a new Scanner object to use (system.in for Keyboard
inputs)
        int num1=0, num2=0, bigger=0; // initialize variables

        System.out.printf("Enter First Number: ");
        num1=input.nextInt(); // store next integer in num1

        System.out.printf("Enter Second Number: ");
        num2=input.nextInt(); // store next integer in num2

        if (num1>num2){ // checks which number is larger
            bigger=num1;
            System.out.printf("%d Is Larger", bigger);
        }
        else if (num1<num2) {
            bigger=num2;
            System.out.printf("%d Is Larger", bigger);
        }
        else { //if both numbers are equal
            System.out.printf("The numbers are equal");
        }
    }
}
```

8. Write an application that inputs three integers from the user and displays the sum, average, product, smallest and largest of the numbers.

```
// File: Question2.java
// Author: Abdulla Faris
import java.util.Scanner; // include scanner utility for accepting keyboard input

public class Question2 { // begin class
    public static void main(String[] args) { // begin the main method
        Scanner input=new Scanner (System.in); //create a new Scanner object to use
        int num1=0, num2=0, num3, bigger=0, smaller=0; // initialize variables
        System.out.printf("NUMBER CALCULATIONS\n\n");

        System.out.printf("Enter First Number:\t\t ");
        num1=input.nextInt(); // store next integer in num1

        System.out.printf("Enter Second Number:\t ");
        num2=input.nextInt(); // store next integer in num2

        System.out.printf("Enter Third Number:\t ");
        num3=input.nextInt(); // store next integer in num3

        bigger=num1>num2?num1:num2; // checks the biggest number in and assigns it to bigger variable
        bigger=bigger>num3?bigger:num3;
        smaller=num1<num2?num1:num2; // checks the smallest number in and assigns it to smaller variable
        smaller=smaller<num3?smaller:num3;

        // display the sum, average, product, smallest and the biggest of all three numbers
        System.out.printf("\t-----\n");
        System.out.printf("\t\t\tSum =\t\t %d\n", num1+num2+num3);
        System.out.printf("\t\t\tAverage =\t %d\n", (num1+num2+num3)/3);
        System.out.printf("\t\t\tProduct =\t %d\n", num1*num2*num3);
        System.out.printf("\t\t\tBiggest =\t %d\n", bigger);
        System.out.printf("\t\t\tSmallest =\t %d\n", smaller);
    }
}
```

9. Write an application that reads two integers, determines whether the first is a multiple of the second and print the result. [Hint Use the remainder operator.]

```
// File: Question3.java
// Author: Abdulla Faris
import java.util.Scanner; // include scanner utility for accepting keyboard input

public class Question3 { // begin class
    public static void main(String[] args) { // begin the main method
        Scanner input=new Scanner (System.in); //create a new Scanner object to use
        int num1=0, num2=0,k; // initialize variables

        System.out.printf("Enter First Number: ");
        num1=input.nextInt(); // store next integer in num1

        System.out.printf("Enter Second Number: ");
        num2=input.nextInt(); // store next integer in num2

        k=num2%num1; // assign the remainder of num2 divided by num1 to the integer k

        if (k==0){ // check if k is 0. remainder k will be 0 if num1 is a multiple of num2,
            System.out.printf("%d is a multiple of %d", num1,num2);
        }
        else {
            System.out.printf("%d is not a multiple of %d", num1,num2);
        }
    }
}
```

10. The process of finding the largest value (i.e., the maximum of a group of values) is used frequently in computer applications. For example, a program that determines the winner of a sales contest would input the number of units sold by each sales person. The sales person who sells the most units wins the contest. Write a Java application that inputs a series of 10 integers and determines and prints the largest integer. Your program should use at least the following three variables:

- counter:** A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed).
- number:** The integer most recently input by the user.
- largest:** The largest number found so far.

```
// File: Question4.java
// Author: Abdulla Faris
import java.util.Scanner; // include scanner utility for accepting keyboard input

public class Question4 { // begin class

    public static void main(String[] args) { // begin the main method
        Scanner input=new Scanner (System.in); //create a new Scanner object to use
        int counter=0, number=0, largest=0; // initialize variables

        for (counter=0; counter<10;counter++){ // loop ten times from 0 to 9
            System.out.printf("Enter Number [%d]: ", counter+1);
            number=input.nextInt(); // store next integer in number
            largest=largest>number?largest:number; // check if new number is larger, if so assign it to larger
        }

        System.out.printf("Largest = %d", largest); // display the largest value
    }
}
```

11. Write a Java application that uses looping to print the following table of values:

| N | 10*N | 100*N | 1000*N |
|---|------|-------|--------|
| 1 | 10   | 100   | 1000   |
| 2 | 20   | 200   | 2000   |
| 3 | 30   | 300   | 3000   |
| 4 | 40   | 400   | 4000   |
| 5 | 50   | 500   | 5000   |

```
// File: Question5.java
// Author: Abdulla Faris
public class Question5 { // begin class

    public static void main(String[] args) { // begin the main method
        int counter=1; // initialize variables
        System.out.printf("N\t10*N\t100*N\t1000*N\n\n", counter, counter*10,
counter*100, counter*1000); // display header
        for (counter=1; counter<=5;counter++){ // loop five times, 1 to 5
            System.out.printf("%d\t%d\t%d\t%d\n", counter, counter*10, counter*100,
counter*1000); // display the table of values
        }
    }
}
```

12. Write a complete Java application to prompt the user for the double radius of a sphere, and call method `sphereVolume` to calculate and display the volume of the sphere. Use the following statement to calculate the volume:

$$\text{double volume} = (4.0 / 3.0) * \text{Math.PI} * \text{Math.pow}(\text{radius}, 3)$$

```
// File: Question6.java
// Author: Abdulla Faris
```

```
import java.util.Scanner; // include scanner utility for accepting keyboard input
```

```

public class Question6 {

    public static double sphereVolume(double radius) { // begin sphereVolume method
        return (4.0/3.0)* Math.PI * Math.pow (radius,3); // return the volume after calculation
    }

    public static void main(String[] args) { // begin the main method
        Scanner input=new Scanner (System.in); //create a new Scanner object to use
        double radius=0.0, volume=0.0; // initialize variables

        System.out.printf("Enter Radius: ");
        radius=input.nextInt();// store next integer in radius

        // display the Volume by calling the sphereVolume method
        System.out.printf("Volume = %.3f", sphereVolume(radius));
    }
}

```

## CHAPTER 4

13. Write statements that perform the following one-dimensional-array operations:

- d. Set the 10 elements of integer array counts to zero.
- e. Add one to each of the 15 elements of integer array bonus.
- f. Display the five values of integer array bestScores in column format.

// File: Question1.java  
 // Author: Abdulla Faris

```

public class Question1 { // begin class

    public static void main(String args[]) { // begin the main method

        // part a
        int array[]={0,0,0,0,0,0,0,0,0,0}; // declaring and setting 10 elements in the array with zero

        // part b
        int bonus[];
        bonus=new int[15]; // declaring array bonus with 15 elements

        for(int i=0;i<15;i++){ // adding 1 to each element
            bonus[i]+=1;
        }

        // part c
        int bestScores[]={10,20,30,40,50}; // declaring the array bestScores of 5 elements
        for (int j=0;j<5;j++){
            System.out.printf("%d\t", bestScores[j]); // displaying them in a column format
        }
    }
}

```

14. Write a Java program that reads a string from the keyboard, and outputs the string twice in a row, first all uppercase and next all lowercase. If, for instance, the string "Hello" is given, the output will be "HELLOhello"

// File: Question2.java  
 // Author: Abdulla Faris

```

import java.util.Scanner; // include scanner utility for accepting input

public class Question2 { // begin class

    public static void main(String[] args) { // begin the main method
        Scanner input=new Scanner(System.in); //create a new Scanner object to use
    }
}

```

```

String str; //declaring a string variable str
System.out.printf("Enter String: ");
str=input.nextLine();// store next line in str

// display the same string in both uppercase and lowercase
System.out.printf("%s%s",str.toUpperCase(),str.toLowerCase());
}
}

```

15. Write a Java application that allows the user to enter up to 20 integer grades into an array. Stop the loop by typing in -1. Your main method should call an Average method that returns the average of the grades. Use the DecimalFormat class to format the average to 2 decimal places.

// File: Question3.java  
 // Author: Abdulla Faris

```

import java.util.Scanner; // include scanner utility for accepting input

public class Question3 { // begin class

    public static double Average(int grades[], int max ) { // begin Average method
        int sum=0; // initialize variables
        double average=0.0;

        for (int i=1;i<max;i++){ // loop and calculate the Average
            sum+=grades[i];
            average=sum/(i);
        }

        return average; // return the average after calculation
    }

    public static void main(String[] args) { // begin the main method

        Scanner input=new Scanner(System.in); //create a new Scanner object to use

        int i, grades[]; // initialize variables
        grades=new int[20];

        for (i=0;i<20;i++){ // start to loop 20 times
            System.out.printf("Enter Grade: ");
            grades[i]=input.nextInt();// store next integer in grades[i]
            if (grades[i]==-1) break;
        }

        System.out.printf("%.2f", Average(grades, i-1));
    }
}

```

## CHAPTER 5

16. Modify class Account (in the example) to provide a method called debit that withdraws money from an Account. Ensure that the debit amount does not exceed the Account's balance. If it does, the balance should be left unchanged and the method should print a message indicating —Debit amount exceeded account balance. Modify class AccountTest (in the example) to test method debit.

```

//filename: Account.java
// Account class
public class Account {
    private double balance;

    public Account(double initialBalance) {

```

```

        if (initialBalance > 0.0) balance=initialBalance;
    }

    public void credit(double amount){
        balance=balance+amount;
    }

    public void debit(double amount){
        balance=balance-amount;
    }

    public double getBalance(){
        return balance;
    }
}

//filename: AccountTest.java
// Account testing class with the main() method

import java.util.Scanner;

public class AccountTest {

    public static void main (String args[]){
        Account account1 = new Account (50.00);
        Account account2 = new Account (-7.53);

        System.out.printf("Account1 Balance: $%.2f\n", account1.getBalance());
        System.out.printf("Account2 Balance: $%.2f\n\n", account2.getBalance());

        Scanner input = new Scanner( System.in );
        double depositAmount;
        double debitAmount;

        System.out.print( "Enter deposit amount for account1: " ); // prompt
        depositAmount = input.nextDouble(); // obtain user input
        System.out.printf( "\nadding %.2f to account1 balance\n\n", depositAmount );
        account1.credit( depositAmount ); // add to account1 balance

        // display balances
        System.out.printf( "Account1 balance: $%.2f\n", account1.getBalance() );
        System.out.printf( "Account2 balance: $%.2f\n\n", account2.getBalance() );

        System.out.print( "Enter deposit amount for account2: " ); // prompt
        depositAmount = input.nextDouble(); // obtain user input
        System.out.printf( "\nAdding %.2f to account2 balance\n\n", depositAmount );
        account2.credit( depositAmount ); // add to account2 balance

        // display balances
        System.out.printf( "Account1 balance: $%.2f\n", account1.getBalance() );
        System.out.printf( "Account2 balance: $%.2f\n", account2.getBalance() );

        System.out.print( "Enter debit amount for account1: " );
        debitAmount = input.nextDouble();
        System.out.printf( "\nSubtracting %.2f from account1 balance\n\n", debitAmount );
        if (account1.getBalance()>=debitAmount) {
            account1.debit( debitAmount );
            System.out.printf( "Account1 balance: $%.2f\n", account1.getBalance() );
            System.out.printf( "Account2 balance: $%.2f\n\n", account2.getBalance() );
        }
        else {
            System.out.printf("!!! Debit amount exceeded account balance!!!\n\n");
        }
    }
}

```



```

// display balances
System.out.print( "Enter debit amount for account2: " );
debitAmount = input.nextDouble();
System.out.printf( "\nSubtracting %.2f from account2 balance\n\n", debitAmount );
if (account1.getBalance()>=debitAmount) {
    account1.debit( debitAmount );
    System.out.printf( "Account1 balance: $%.2f\n", account1.getBalance() );
    System.out.printf( "Account2 balance: $%.2f\n\n", account2.getBalance() );
}
else {
    System.out.printf("!!!Debit amount exceeded account balance!!!\n\n");
}
}
}

```

17. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables—a part number(type String),a part description(type String),a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoice Amount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test application named InvoiceTest that demonstrates class Invoice’s capabilities.

```

//filename: Invoice.java
// Invoice class
public class Invoice {
    private String partNumber;
    private String partDescription;
    private int quantity;
    private double price;

    public Invoice(String pNum, String pDesc, int qty, double prc) {
        if (pNum != null) partNumber=pNum; else partNumber="0";
        if (pDesc != null) partDescription=pDesc; else partDescription="0";
        if (qty > 0) quantity=qty; else quantity=0;
        if (prc > 0.0) price=prc; else price=0;
    }

    public String getPartNum(){
        return partNumber;
    }

    public String getPartDesc(){
        return partDescription;
    }

    public int getQuantity(){
        return quantity;
    }

    public double getPrice(){
        return price;
    }

    public void setPartNum(String pNum){
        if (pNum != null) {partNumber=pNum;}
        else {partNumber="0";}
    }

    public void setPartDesc(String pDesc){
        if (pDesc != null) {partDescription=pDesc;}
        else {partDescription="0";}
    }
}

```

```

public void setQuantity(int qty){
    if (qty > 0) {quantity=qty;}
    else {quantity=0;}
}

public void setPrice(double prc){
    if (prc > 0.0) {price=prc;}
    else {price=0.0;}
}

public double getInvoiceAmount(){
    return (double)quantity*price;
}
}

```

//filename: InvoiceTest.java  
// Invoice testing class with the main() method

```

public class InvoiceTest {
    public static void main (String args[]){
        Invoice invoice1=new Invoice ("A5544", "Big Black Book", 500, 250.00);
        Invoice invoice2=new Invoice ("A5542", "Big Pink Book", 300, 50.00);

        System.out.printf("Invoice 1: %s\t%s\t%d\t$%.2f\n", invoice1.getPartNum(),
invoice1.getPartDesc(), invoice1.getQuantity(), invoice1.getPrice());
        System.out.printf("Invoice 2: %s\t%s\t%d\t$%.2f\n", invoice2.getPartNum(),
invoice2.getPartDesc(), invoice2.getQuantity(), invoice2.getPrice());
    }
}

```

18. Create a class called Employee that includes three pieces of information as instance variables—a first name (typeString), a last name (typeString) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.

```

//filename: Employee.java
// Employee class
public class Employee {
    private String firstName;
    private String lastName;
    private double salary;

    public Employee(String fName, String lName, double sal) {
        if (fName != null) firstName =fName;
        if (lName != null) lastName = lName;
        if (sal > 0.0) {
            salary=sal;
        }
        else {
            salary=0.0;
        }
    }

    //set methods
    public String getFirstName(){
        return firstName;
    }
    public String getLastName(){
        return lastName;
    }
    public double getSalary(){
        return salary;
    }
}

```



```

        month = myMonth;
        day = myDay;
        year = myYear;
    }

    public void setMonthDate(int myMonth) {
        month = myMonth;
    }

    public int getMonthDate() {
        return month;
    }

    public void setDayDate(int myDay) {
        day = myDay;
    }

    public int getDayDate() {
        return month;
    }

    public void setYearDate(int myYear) {
        year = myYear;
    }

    public int getYearDate() {
        return year;
    }

    public void displayDate() {
        System.out.printf("%d/%d/%d", month, day, year);
    }
}

//filename: DateTest.java
// Date testing class with the main() method

import java.util.*;
public class DateTest {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Date myDate = new Date(9, 11, 1986);

        System.out.println("Enter The Month");
        int myMonth = input.nextInt();
        myDate.setMonthDate(myMonth);

        System.out.println("Enter the Date");
        int myDay = input.nextInt();
        myDate.setDayDate(myDay);

        System.out.println("Enter the Year");
        int myYear = input.nextInt();
        myDate.setYearDate(myYear);

        myDate.displayDate();
    }
}

```

## CHAPTER 6

20. Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly

interest by multiplying the savingsBalance by annualInterestRate divided by 12 this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of \$2000.00 and \$3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.

```
//filename: SavingsAccount.java
// SavingsAccount class
public class SavingsAccount {
    public static double annualInterestRate;
    private double savingsBalance;

    public SavingsAccount() {
        annualInterestRate = 0.0;
        savingsBalance = 0.0;
    }

    public SavingsAccount(double intRate, double savBal) {
        annualInterestRate = intRate;
        savingsBalance = savBal;
    }

    public double calculateMonthlyInterest() {
        double intRate = (savingsBalance * annualInterestRate/12);
        savingsBalance = savingsBalance + intRate;
        return intRate;
    }

    public static void modifyInterestRate(double newInterestRate) {
        annualInterestRate = newInterestRate;
    }

    public void setSavingsBalance(double newBal) {
        savingsBalance = newBal;
    }

    public double getSavingsBalance() {
        return savingsBalance;
    }

    public double getAnnualInterestRate() {
        return annualInterestRate;
    }
}
```

```
//filename: SavingsAccountTest.java
// SavingsAccount testing class with the main() method

public class SavingsAccountTest {

    public static void main(String[] args) {
        SavingsAccount saver1 = new SavingsAccount();
        SavingsAccount saver2 = new SavingsAccount();

        saver1.setSavingsBalance(2000.00);
        saver2.setSavingsBalance(3000.00);

        SavingsAccount.modifyInterestRate(0.04);
        saver1.calculateMonthlyInterest();
        saver2.calculateMonthlyInterest();

        System.out.printf("New Balance for Saver1=%f\n",saver1.getSavingsBalance());
        System.out.printf("New Balance for Saver2=%f\n",saver2.getSavingsBalance());
    }
}
```

```

        SavingsAccount.modifyInterestRate(0.05);
        saver1.calculateMonthlyInterest();
        saver2.calculateMonthlyInterest();

        System.out.printf("New Balance for Saver1=%f\n",saver1.getSavingsBalance());
        System.out.printf("New Balance for Saver2=%f\n",saver2.getSavingsBalance());
    }
}

```

21. Create a class called Book to represent a book. A Book should include four pieces of information as instance variables-a book name, an ISBN number, an author name and a publisher. Your class should have a constructor that initializes the four instance variables. Provide a mutator method and accessor method (query method) for each instance variable. In addition, provide a method named getBookInfo that returns the description of the book as a String (the description should include all the information about the book). You should use this keyword in member methods and constructor. Write a test application named BookTest to create an array of object for 30 elements for class Book to demonstrate the class Book's capabilities.

```

//filename: Book.java
// Book class
public class Book {
    private String Name;
    private String ISBN;
    private String Author;
    private String Publisher;

    public Book() {
        Name = "NULL";
        ISBN = "NULL";
        Author = "NULL";
        Publisher = "NULL";
    }

    public Book(String name, String isbn, String author, String publisher) {
        Name = name;
        ISBN = isbn;
        Author = author;
        Publisher = publisher;
    }

    public void setName(String Name) {
        this.Name = Name;
    }
    public String getName() {
        return Name;
    }
    public void setISBN(String ISBN) {
        this.ISBN = ISBN;
    }
    public String getISBN() {
        return ISBN;
    }
    public void setAuthor(String Author) {
        this.Author = Author;
    }
    public String getAuthor() {
        return Author;
    }
    public void setPublisher(String Publisher) {
        this.Publisher = Publisher;
    }
    public String getPublisher() {
        return Publisher;
    }
}

```

```

    public void getBookInfo() {
        System.out.printf("%s %s %s %s", Name,ISBN,Author,Publisher);
    }
}

//filename: SavingsAccountTest.java
// SavingsAccount testing class with the main() method

public class BookTest {

    public static void main(String[] args) {

        Book test[] = new Book[13];
        test[1] = new Book();
        test[1].getBookInfo();

    }
}

```

## CHAPTER 7

- 22.
- Create a super class called **Car**. The Car class has the following fields and methods.
    - **int** speed;
    - **double** regularPrice;
    - **String** color;
    - **double** getSalePrice();

```

//filename: Car.java
//Car class
public class Car {
    private int speed;
    private double regularPrice;
    private String color;

    public Car (int Speed,double regularPrice,String color) {
        this.speed = Speed;
        this.regularPrice = regularPrice;
        this.color = color;
    }

    public double getSalePrice() {
        return regularPrice;
    }
}

```

- Create a sub class of Car class and name it as **Truck**. The Truck class has the following fields and methods.
  - **int** weight;
  - **double** getSalePrice();//If weight>2000,10%discount.Otherwise,20%discount.

```

//filename: Truck.java
// Truck class, subclass of Car
public class Truck extends Car {
    private int weight;

    public Truck (int Speed,double regularPrice,String color, int weight) {
        super(Speed,regularPrice,color);
        this.weight = weight;
    }
}

```

```

    public double getSalePrice() {
        if (weight > 2000){
            return super.getSalePrice() - (0.1 * super.getSalePrice());
        }
        else {
            return super.getSalePrice();
        }
    }
}

```

c. Create a subclass of Car class and name it as **Ford**. The Ford class has the following fields and methods

- **int** year;
- **int** manufacturerDiscount;
- **double** getSalePrice(); // From the sale price computed from Car class, subtract the manufacturerDiscount.

```

//filename: Ford.java
// Ford class, subclass of Car
public class Ford extends Car {
    private int year;
    private int manufacturerDiscount;

    public Ford (int Speed, double regularPrice, String color, int year, int
manufacturerDiscount) {
        super (Speed, regularPrice, color);
        this.year = year;
        this.manufacturerDiscount = manufacturerDiscount;
    }

    public double getSalePrice() {
        return (super.getSalePrice() - manufacturerDiscount);
    }
}

```

d. Create a subclass of Car class and name it as **Sedan**. The Sedan class has the following fields and methods.

- **int** length;
- **double** getSalePrice(); // If length > 20 feet, 5% discount, Otherwise, 10% discount.

```

//filename: Sedan.java
// Sedan class, subclass of Car
public class Sedan extends Car {
    private int length;

    public Sedan (int Speed, double regularPrice, String color, int length) {
        super (Speed, regularPrice, color);
        this.length = length;
    }

    public double getSalePrice() {
        if (length > 20) {
            return super.getSalePrice() - (0.05 * super.getSalePrice());
        }
        else {
            return super.getSalePrice() - (0.1 * super.getSalePrice());
        }
    }
}

```

e. Create **MyOwnAutoShop** class which contains the main() method. Perform the following within the main() method.

- Create an instance of Sedan class and initialize all the fields with appropriate values. Use **super(...)** method in the constructor for initializing the fields of the superclass.
- Create two instances of the Ford class and initialize all the fields with appropriate values. Use **super(...)** method in the constructor for initializing the fields of the super class.



- Create an instance of Car class and initialize all the fields with appropriate values.  
Display the sale prices of all instance.

```
//filename: MyOwnAutoShop.java
// Testing class with the main() method
public class MyOwnAutoShop {
    (int Speed, double regularPrice, String color, int year, int manufacturerDiscount)

    public static void main(String[] args) {
        Sedan mySedan = new Sedan(160, 20000, "Red", 10);
        Ford myFord1 = new Ford (156, 4452.0, "Black", 2005, 10);
        Ford myFord2 = new Ford (155, 5000.0, "Pink", 1998, 5);
        Car myCar = new Car (555, 56856.0, "Red");

        System.out.printf("MySedan Price %.2f", mySedan.getSalePrice());
        System.out.printf("MyFord1 Price %.2f", myFord1.getSalePrice());
        System.out.printf("MyFord2 Price %.2f", myFord2.getSalePrice());
        System.out.printf("MyCar Price %.2f", myCar.getSalePrice());

    }
}
```

## CHAPTER 8

## CHAPTER 9

23. Write an applet that asks the user to enter two floating-point numbers, obtains the two numbers from the user and draws their sum, product (multiplication), difference and quotient (division). Use the techniques shown in example.

```
//filename: simpleApplet1.java

import java.awt.*;
import javax.swing.*;

public class simpleApplet1 extends JApplet {
    double sum;
    double product;
    double difference;
    double quotient;

    public void init() {
        String firstNumber;
        String secondNumber;

        double number1;
        double number2;

        firstNumber = JOptionPane.showInputDialog("Enter the first number");
        secondNumber = JOptionPane.showInputDialog("Enter the second number");

        number1 = Double.parseDouble(firstNumber);
        number2 = Double.parseDouble(secondNumber);

        sum = number1 + number2;
        product = number1 * number2;
        difference = number1 - number2;
        quotient = number1 / number2;

    }

    public void paint(Graphics g) {
```

```

        super.paint(g);
        g.drawRect(15, 10, 270, 60);

        g.drawString("Sum          "+sum, 25, 25);
        g.drawString("Product        "+product, 25, 35);
        g.drawString("Difference      "+difference, 25, 45);
        g.drawString("Quotient       "+quotient, 25, 55);
    }

}

```

## CHAPTER 10

**24. Create an applet that can display the following component. No event handling is needed for the components.**

//filename: simpleAppet2.java

```

import javax.swing.*.*;
import java.awt.*.*;
public class simpleAppet2 extends JApplet {
    private JLabel lblName;
    private JLabel lblAddress;
    private JLabel lblEmail;
    private JTextField txtName;
    private JTextField txtAddress;
    private JTextField txtEmail;

    public void init() {
        Container conpane = getContentPane();
        conpane.setLayout(new FlowLayout());
        lblName = new JLabel("name");
        lblAddress = new JLabel("address");
        lblEmail = new JLabel("email");

        txtName = new JTextField(10);
        txtAddress = new JTextField(10);
        txtEmail = new JTextField(10);

        conpane.add(lblName);
        conpane.add(txtName);
        conpane.add(lblAddress);
        conpane.add(txtAddress);
        conpane.add(lblEmail);
        conpane.add(txtEmail);
    }
}

```

**25. Create an applet that can display the following component. No event handling is needed for the components.**

//filename: simpleAppet3.java

```

import javax.swing.*.*;
import java.awt.*.*;

public class simpleAppet3 extends JApplet {
    private JButton button1, button2, button3, button4, button5;

    public void init() {
        Container conpane = getContentPane();
        conpane.setLayout(new BorderLayout());
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(1, 5));
    }
}

```



```

        button1 = new JButton("Button 1");
        button2 = new JButton("Button 2");
        button3 = new JButton("Button 3");
        button4 = new JButton("Button 4");
        button5 = new JButton("Button 5");

        panel.add(button1);
        panel.add(button2);
        panel.add(button3);
        panel.add(button4);
        panel.add(button5);

        conpane.add("South",panel);

    }
}

```

## CHAPTER 11

### 26. Temperature Conversion

- Write a temperature conversion applet that converts from Fahrenheit to Celsius. The Fahrenheit temperature should be entered from the keyboard (via a JTextField). A JLabel should be used to display the converted temperature. Use the following formula for the conversion:  
**Celcius = ((5/9)\*(Ferenheit-32)).**
- Enhance the temperature conversion applet of Q1 by adding the Kelvin temperature scale. The applet should also allow the user to make conversions between any two scales. Use the following formula for the conversion between Kelvin and Celsius (in addition to the formulain Q1): **Kelvin = Celcius + 273.15**

```

/*
 * Filename: tempCon.java
 * Author: Abdulla Faris
 * Date: 13/04/2010
 */

import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;
import java.text.*;

```

```

public class tempCon extends JApplet implements ActionListener {

    JTextField txtInput;
    JLabel lblResult;
    JRadioButton rbCelcius, rbKelvin;

    public void init(){

        Container conpane = getContentPane();
        conpane.setLayout (new FlowLayout());

        txtInput = new JTextField("",10);
        conpane.add(txtInput);

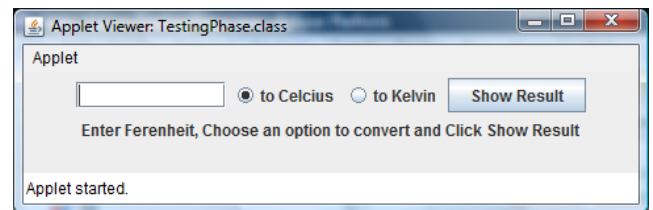
        rbCelcius= new JRadioButton ("to Celcius", true);
        conpane.add(rbCelcius);

        rbKelvin = new JRadioButton("to Kelvin", false);
        conpane.add(rbKelvin);

        ButtonGroup selection = new ButtonGroup();
        selection.add(rbCelcius);
        selection.add(rbKelvin);

        JButton button1 = new JButton ("Show Result");
        button1.addActionListener(this);
        conpane.add(button1);
    }
}

```



```

        lblResult= new JLabel ("Enter Ferenheit, Choose an option to convert and Click
Show Result");
        conpane.add(lblResult);
    }

    public void actionPerformed(ActionEvent e) {

        DecimalFormat df = new DecimalFormat ("#.##");
        double ferenheit = Double.parseDouble(txtInput.getText());
        double answer = 0.0;

        answer = ((5.0/9.0)*(ferenheit - 32.0));

        if (rbKelvin.isSelected())
            answer += 273.15;

        lblResult.setText(String.valueOf(df.format(answer)));
    }
}

```