Exercises

1. (Keyword new) What's the purpose of keyword new? Explain what happens when you use it.
   My answer
   1.1. The keyword new allows you to create a new data type that is needed for use in a Java application.
   1.2. When one uses the new keyword, a new object is created, which can be used in the Java application by utilizing the methods that are available within that object.
2. (Default Constructors) What is a default constructor? How are an object's instance variables initialized if a class has only a default constructor?
   My answer
   1.1. A default constructor is a constructor that is provided by the compiler if one has not been created.
   1.2. The instance variables are automatically initialized to a value of null due to the default constructor that is provided by the compiler.
3. (Instance Variables) Explain the purpose of an instance variable.
   My answer
   3.1. An instance variable serves as an attribute of an object that would have been created and its methods used to manipulate the objects attribute.
4. (Using Classes without Importing Them) Most classes need to be imported before they can be used in an app. Why is every app allowed to use classes System and String without first importing them?
   My answer
   4.1. Both class System and String are part of the java.lang package, which is automatically imported in the Java application, meaning that it does not require explicit import.
5. (Using a Class without Importing It) Explain how a program could use class Scanner without importing it.
   My answer
   5.1. Scanner can be used by quoting the full package name and method like this: java.util.Scanner everytime that it is needed for use in the Java application.
6. (set and get Methods) Explain why a class might provide a set method and a get method for an instance variable.
   My answer
   6.1. A class would need set and get methods for an instance variable to be able enable proper manipulation of the instance variable and to avoid the instance variable being misused or corrupted by a malicious user or programmer.
7. (Modified Account Class) Modify class Account (Fig. 3.8) to provide a method called withdraw that withdraws money from an Account. Ensure that the withdrawal 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 "Withdrawal amount exceeded account balance." Modify class AccountTest (Fig. 3.9) to test method withdraw.

My answer

```
//Exercise 3.11
//Program inputs strings and integers and out puts account name and
//account balance

public class Account311
{
    private String name;
    private double balance;

    public Account311(String name,double amount)
    {
        this.name=name;

        if(amount>0.0)
        {
            this.balance=amount;
        }
    }

    public void deposit(double depositAmount)
    {
        if(depositAmount>0.0)
        {
            this.balance=depositAmount;
        }
    }

    public void withdrawal(double widthdrawAmount)
    {
        balance=balance-widthdrawAmount;

        if(balance<=0.0)
        {
            balance=0.0;
        }
    }
```

```
    public double getDeposit()
    {
        if(balance==0.00)
        {
            System.out.println("Account Withdrawal exceeded.");
        }
        return balance;
    }

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

    public String getName()
    {
        return name;
    }
}

//Exercise 3.11
//Program inputs strings and integers and out puts account name and
//account balance

import java.util.Scanner;   //Program uses class Scanner

public class AccountTest311
{
    //method main begins execution of Java application
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);
        Account311 account1=new Account311("Njabulo",500);
        Account311 account2=new Account311("Sindiswa",850);

        System.out.printf("Account1 information:%s R%.2f%n",account1.getName(),account1.getDeposit());
        System.out.printf("Account2 information:%s R%.2f%n",account2.getName(),account2.getDeposit());

        System.out.print("Enter withdrawal Amount:");
        double rands=input.nextDouble();

        account1.withdrawal(rands);

        System.out.printf("%nAccount1 information:%s R%.2f%n",account1.getName(),account1.getDeposit());
        System.out.printf("Account2 information:%s R%.2f%n",account2.getName(),account2.getDeposit());
    }
}
```

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RORI>cd/Users/RORI/Desktop/Learning Java/ch3/Exercises/Exercise3.11

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.11>javac *.java

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.11>java Exercise311
Error: Could not find or load main class Exercise311

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.11>java AccountTest311
Account1 information:Njabulo R500.00
Account2 information:Sindiswa R850.00
Enter withdrawal Amount:1000
Account Withdrawal exceeded.

Account1 information:Njabulo R0.00
Account2 information:Sindiswa R850.00

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.11>
```

8. (Invoice Class) 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 getInvoiceAmount 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 app named InvoiceTest that demonstrates class Invoice's capabilities
   My answer

```java
//Exercise 3.12
//Program prints out an invoice based on information entered by the
//user

public class Account312
{
    private String partNumber;
    private String partDescription;
    private int partQuantity;
    private double partPrice;

    public Account312(String number,String description,int quantity,double price)
    {
        this.partNumber=number;

        this.partDescription=description;

        if(quantity>0)
        {
            this.partQuantity=quantity;
        }

        if(price>0.0)
        {
            this.partPrice=price;
        }
    }

    public void setNumber(String number)
    {
        this.partNumber=number;
    }
```

```
    public String getNumber()
    {
        return partNumber;
    }

    public void setDescription(String description)
    {
        this.partDescription=description;
    }

    public String getDescription()
    {
        return partDescription;
    }

    public void setQuantity(int quantity)
    {
        this.partQuantity=quantity;
    }

    public int getQuantity()
    {
        if(partQuantity<=0)
        {
            partQuantity=0;
        }
        return partQuantity;
    }

    public void setPrice(double price)
    {
        this.partPrice=price;
    }

    public double getPrice()
    {
        if(partPrice<=0.0)
        {
            partPrice=0.0;
        }
        return partPrice;
    }

    public double getInvoiceAmount(double price,int quantity)
    {
        if(partPrice>0.0)
        {
            partPrice=price*(double)quantity;
        }
        return partPrice;
    }
}
```

```java
//Exercise 3.12
//Program prints out an invoice based on information entered by the
//user

import java.util.Scanner;   //program uses class Scanner

public class AccountTest312
{
    //method main bejins execution of Java application
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);

        Account312 account1=new Account312("5010","DiskDrive",1,100.0);
        Account312 account2=new Account312("5020","DiskTray",5,250.0);

        System.out.printf("Hardware parts for account1:%n%s%n%s%n%d%n%.2f%n",
        account1.getNumber(),account1.getDescription(),account1.getQuantity(),
        account1.getPrice());

        System.out.printf("%nHardware parts for account2:%n%s%n%s%n%d%n%.2f%n",
        account2.getNumber(),account2.getDescription(),account2.getQuantity(),
        account2.getPrice());

        System.out.print("Enter partNumber:");
        String number=input.nextLine();

        System.out.print("Enter partDescription:");
        String description=input.nextLine();

        System.out.print("Enter partQuantity:");
        int quantity=input.nextInt();

        System.out.print("Enter partPrice:");
        double price=input.nextDouble();

        account1.setNumber(number);
        account1.setDescription(description);
        account1.setQuantity(quantity);
        account1.setPrice(price);
        account1.getInvoiceAmount(price,quantity);

        System.out.printf("Hardware parts for account1:%n%s%n%s%n%d%n%.2f%n",
        account1.getNumber(),account1.getDescription(),account1.getQuantity(),
        account1.getPrice());

        System.out.printf("Hardware parts for account2:%n%s%n%s%n%d%n%.2f%n",
        account2.getNumber(),account2.getDescription(),account2.getQuantity(),
        account2.getPrice());
    }
}
```

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RORI>cd/USers/RORI/Desktop/Learning Java/ch3/Exercises/Exercise3.12

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.12>javac *.java

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.12>java AccountTest312
Hardware parts for account1:
5010
DiskDrive
1
100.00

Hardware parts for account2:
5020
DiskTray
5
250.00
Enter partNumber:5046
Enter partDescription:GPU
Enter partQuantity:5
Enter partPrice:6000
Hardware parts for account1:
5046
GPU
5
30000.00
Hardware parts for account2:
5020
DiskTray
5
250.00

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.12>javac AccountTest312
error: Class names, 'AccountTest312', are only accepted if annotation processing is explicitly requested
1 error

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.12>java AccountTest312
Hardware parts for account1:
5010
DiskDrive
1
100.00

Hardware parts for account2:
5020
DiskTray
5
250.00
Enter partNumber:5060
Enter partDescription:CPU
Enter partQuantity:-3
Enter partPrice:-5
Hardware parts for account1:
5060
CPU
0
0.00
Hardware parts for account2:
```

```
5020
DiskTray
5
250.00

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.12>
```

9. (Employee Class) Create a class called Employee that includes three instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Provide 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, do not set its value. Write a test app 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.

My answer

```java
//Exercise 3.13
//Program takes input from the user and outputs the name surname and
//salary of the employee pluse adds 10% bonus

public class Account313
{
    private String employeeName;
    private String employeeSurname;
    private double employeeSalary;

    public Account313(String name,String surname,double salary)
    {
        this.employeeName=name;

        this.employeeSurname=surname;

        if(salary>0.0)
        {
            this.employeeSalary=salary;
        }
    }

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

    public String getName()
    {
        return employeeName;
    }

    public void setSurname(String surname)
    {
        this.employeeSurname=surname;
    }
```

```
    public String getSurname()
    {
        return employeeSurname;
    }

    public void setSalary(double salary)
    {
        if(salary>0.0)
        {
            employeeSalary=((employeeSalary+(salary*10))/100)+salary;
        }
    }

    public double getSalary()
    {
        return employeeSalary;
    }
}
```

```java
//Exercise 3.13
//Program takes input from the user and outputs the name surname and
//salary of the employee pluse adds 10% bonus

import java.util.Scanner;   //program uses class Scanner;

public class AccountTest313
{
    //method main begins execution of Java application
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);

        Account313 account1=new Account313("Clyton","Makasi",3500);
        Account313 account2=new Account313("Lelethu","Makasi",9500);

        System.out.printf("Employee account1 information: %s %s %.2f %n",
        account1.getName(),account1.getSurname(),account1.getSalary());

        System.out.printf("%nEmployee account1 information: %s %s %.2f %n",
        account2.getName(),account2.getSurname(),account2.getSalary());

        System.out.print("Enter new amount for account1:");
        double increase=input.nextDouble();
        account1.setSalary(increase);

        System.out.print("Enter new amount for account2:");
        increase=input.nextDouble();
        account2.setSalary(increase);

        System.out.printf("%nEmployee account1 information: %s %s %.2f %n",
        account1.getName(),account1.getSurname(),account1.getSalary());

        System.out.printf("%nEmployee account1 information: %s %s %.2f %n",
        account2.getName(),account2.getSurname(),account2.getSalary());
    }
}
```

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RORI>cd/Users/RORI/Desktop/Learning Java/ch3/Exercises/Exercise3.13

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.13>javac *.java

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.13>java AccountTest313
Employee account1 information: Clyton Makasi 3500.00

Employee account1 information: Lelethu Makasi 9500.00
Enter new amount for account1:6000
Enter new amount for account2:10000

Employee account1 information: Clyton Makasi 6635.00

Employee account1 information: Lelethu Makasi 11095.00

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.13>
```

10. (Date Class) Create a class called Date that includes three instance variables—a month (type int), a day (type int) and a year (type int). Provide a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test app named DateTest that demonstrates class Date's capabilities.

My answer

```
//Exercise 3.14
//Program input s inegers and outputs current date

public class Account314
{
    private int currentMonth;
    private int currentDay;
    private int currentYear;

    public Account314(int month,int day,int year)
    {
        if(month>0)
        {
            this.currentMonth=month;
        }

        if(day>0)
        {
            this.currentDay=day;
        }

        if(year>0)
        {
            this.currentYear=year;
        }
    }

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

    public int getMonth()
    {
        return currentMonth;
    }
```

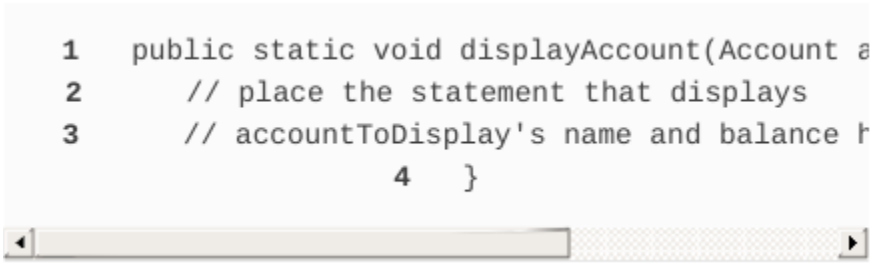
```
    public void setDay(int day)
    {
        this.currentDay=day;
    }

    public int getDay()
    {
        return currentDay;
    }

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

    public int getYear()
    {
        return currentYear;
    }
}
```

```java
//Exercise 3.14
//Program input s inegers and outputs current date

import java.util.Scanner;   //program uses class Scanner

public class AccountTest314
{
    //method main begins execution of Java application
    public static void main(String[] args)
    {
        Scanner input=new Scanner(System.in);

        Account314 account1=new Account314(5,10,1965);

        System.out.printf("Currrent date is: %d / %d / %d%n",
        account1.getMonth(),account1.getDay(),account1.getYear());

        System.out.print("Enter a month between 1 and 12:");
        int month=input.nextInt();
        account1.setMonth(month);

        System.out.print("Enter a day between 1 and 31:");
        int day=input.nextInt();
        account1.setDay(day);

        System.out.print("Enter a year:");
        int year=input.nextInt();
        account1.setYear(year);

        System.out.printf("Currrent date is: %d / %d / %d%n",
        account1.getMonth(),account1.getDay(),account1.getYear());
    }
}
```

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RORI>cd/Users/RORI/Desktop/Learning Java/ch3/Exercises/Exercise3.14

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.14>javac *.java

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.14>java AccountTest314
Currrent date is: 5 / 10 / 1965
Enter a month between 1 and 12:6
Enter a day between 1 and 31:25
Enter a year:2015
Currrent date is: 6 / 25 / 2015

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.14>
```

11. (Removing Duplicated Code in Method main) In the AccountTest class of Fig. 3.9 , method main contains six statements (lines 11–12, 13–14, 26–27, 28–29, 38–39 and 40–41) that each display an Account object's name and balance. Study these statements

and you'll notice that they differ only in the Account object being manipulated—account1 or account2. In this exercise, you'll define a new displayAccount method that contains one copy of that output statement. The method's parameter will be an Account object and the method will output the object's name and balance. You'll then replace the six duplicated statements in main with calls to displayAccount, passing as an argument the specific Account object to output. Modify class AccountTest of Fig. 3.9 to declare method displayAccount (Fig. 3.20) after the closing right brace of main and before the closing right brace of class AccountTest. Replace the comment in the method's body with a statement that displays accountToDisplay's name and balance.



Method displayAccount to add to class Account. Recall that main is a static method, so it can be called without first creating an object of the class in which main is declared. We also declared method displayAccount as a static method. When main needs to call another method in the same class without first creating an object of that class, the other method also must be declared static. Once you've completed displayAccount's declaration, modify main to replace the statements that display each Account's name and balance with calls to displayAccount—each receiving as its argument the account1 or account2 object, as appropriate. Then, test the updated AccountTest class to ensure that it produces the same output as shown in Fig. 3.9 .

My answer

```java
//Exercise 315
//An upgraded version of the Account app that shows the information
//without having diplicated code.

public class Account315
{
    private String name;
    private double balance;

    public Account315(String name,double amount)
    {
        this.name=name;

        if(amount>0.0)
        {
            this.balance=amount;
        }
    }

    public void deposit(double depositAmount)
    {
        if(depositAmount>0.0)
        {
            this.balance=depositAmount;
        }
    }

    public void withdrawal(double widthdrawAmount)
    {
        balance=balance-widthdrawAmount;

        if(balance<=0.0)
        {
            balance=0.0;
        }
    }
```

```
    public double getDeposit()
    {
        if(balance==0.00)
        {
            System.out.println("Account Withdrawal exceeded.");
        }
        return balance;
    }

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

    public String getName()
    {
        return name;
    }
}
```

```java
//Exercise 315
//An upgraded version of the Account app that shows the information
//without having diplicated code.

import java.util.Scanner;   //program uses class scanner

public class AccountTest315
{
    //method main begins execution of Java application
    public static void main(String[] args)
    {
        Account315 account1=new Account315("Jane Green ",50.00);
        Account315 account2=new Account315("John Blue",0.00);

        //display initial balance of each object
        displayAccount(account1);
        displayAccount(account2);

        //create a Scanner to obtain input from the keyboard
        Scanner input=new Scanner(System.in);

        System.out.print("Enter deposit amount for account1:");
        double depositAmount=input.nextDouble();

        System.out.printf("%nAdding %.2f to account1 balance%n",depositAmount);
        account1.deposit(depositAmount);

        //display balances
        displayAccount(account1);
        displayAccount(account2);

        System.out.print("Enter deposit amount for account2:");
        depositAmount=input.nextDouble();

        System.out.printf("%nAdding %.2f to account2 balance%n",depositAmount);
        account2.deposit(depositAmount);

        //display balances
        displayAccount(account1);
        displayAccount(account2);
    }

    public static void displayAccount(Account315 current)
    {
        System.out.printf("%s balance: $%.2f%n",current.getName(),current.getDeposit());
    }
}
```

```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RORI>cd/Users/RORI/Desktop/Learning Java/ch3/Exercises/Exercise3.15

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.15>javac *.java

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.15>java AccountTest315
Jane Green  balance: $50.00
Account Withdrawal exceeded.
John Blue balance: $0.00
Enter deposit amount for account1:1000

Adding 1000.00 to account1 balance
Jane Green  balance: $1000.00
Account Withdrawal exceeded.
John Blue balance: $0.00
Enter deposit amount for account2:1500

Adding 1500.00 to account2 balance
Jane Green  balance: $1000.00
John Blue balance: $1500.00

C:\Users\RORI\Desktop\Learning Java\ch3\Exercises\Exercise3.15>
```