Exercise 1 (Packages) Your creativity

1. Create three packages, think of them yourself
2. Put two different classes in each package
3. Import all three of these packages in class named PackagePractice, you will have access to 6 classes
4. Call methods of these 6 classes and use them in PackagePractice
5. package Package1;  
     
   public class Employee {  
    private String firstName;  
    private String lastName;  
    private String nationalIdCardNumber;  
    public Employee () {  
    }  
    public Employee(String firstName, String lastName, String nationalIdCardNumber) {  
    this.firstName = firstName;  
    this.lastName = lastName;  
    this.nationalIdCardNumber = nationalIdCardNumber;  
    }  
    void setFirstName(String firstName){  
    this.firstName = firstName;  
    }  
    String getFirstName(){  
    return firstName;  
    }  
    void setLastName(String lastName){  
    this.lastName = lastName;  
    }  
    String getLastName(){  
    return lastName;  
    }  
    void setNationalIdCardNumber(String nationalIdCardNumber){  
    this.nationalIdCardNumber = nationalIdCardNumber;  
    }  
    String getNationalIdCardNumber(){  
    return nationalIdCardNumber;  
    }  
    public String tostring() {  
    return firstName+ " " + lastName + "CNIC" + nationalIdCardNumber;  
    }  
    double earnings(){  
    return 0.0;  
    }  
   }
6. package Package1;  
     
   public class SalariedEmployee extends Employee {  
    private double weeklySalaried;  
    SalariedEmployee(){  
     
    }  
    public SalariedEmployee(String firstName, String lastName, String nationalIdCardNumber, double weeklySalaried) {  
    super(firstName, lastName, nationalIdCardNumber);  
    this.weeklySalaried = weeklySalaried;  
    }  
    void setWeeklySalaried(double weeklySalaried) {  
    this.weeklySalaried = weeklySalaried;  
    }  
    double weeklySalaried(double weeklySalaried) {  
    return weeklySalaried;  
    }  
    public String tostring(){  
    return super.tostring() + " " + weeklySalaried;  
    }  
    public double earnings() {  
    if (weeklySalaried >= 0.0){  
    return weeklySalaried;  
    }  
    else {  
    return 0.0;  
    }  
    }  
     
   }
7. package Package2;  
   import Package1.\*;  
   public class CommissionEmployee extends Employee {  
    private double grossSales;  
    private double commissionRate;  
    public CommissionEmployee() {}  
    public CommissionEmployee (String firstName, String lastName, String nationalIdCardNumber, double grossSales, double commissionRate) {  
    super(firstName, lastName, nationalIdCardNumber);  
    this.grossSales = grossSales;  
    this.commissionRate = commissionRate;  
    }  
    void setGrossSales(double grossSales) {  
    this.grossSales = grossSales;  
    }  
    double getGrossSales() {  
    return grossSales;  
    }  
    void setCommissionRate(double grossSales) {  
    this.grossSales = grossSales;  
    }  
    double getCommissionRate() {  
    return grossSales;  
    }  
    public String tostring() {  
    return "\nCommission employee: " + super.tostring();  
    }  
    public double earnings() {  
    return grossSales \* commissionRate;  
    }  
     
   }
8. package Package2;  
   import Package1.\*;  
     
   public class HourlyEmployee extends Employee {  
    private double wage;  
    private int hours;  
    public HourlyEmployee() {  
    super();  
    }  
    public HourlyEmployee(String firstName, String lastName, String nationalIdCardNumber, double wage, int hour){  
    super(firstName, lastName, nationalIdCardNumber);  
    if (wage >= 0) {  
    this.wage = wage;  
    }  
    else {  
    this.wage = 0.0;  
    }  
    if (hours >= 0) {  
    this.hours = hours;  
    }  
    else {  
    this.hours = 0;  
    }  
    }  
    void setWage(double wage) {  
    if (wage >= 0) {  
    this.wage = wage;  
    }  
    else {  
    this.wage = 0.0;  
    }  
    }  
    double getWage(){  
    return wage;  
    }  
    void setHours(int hours) {  
    if (hours >= 0) {  
    this.hours = hours;  
    }  
    else {  
    this.hours = 0;  
    }  
    }  
    double getHours(){  
    return hours;  
    }  
    public double earnings( ) {  
    if (hours <= 40){  
    return wage \* hours;  
    }  
    else {  
    return 40\*wage + (hours-40)\*wage\*1.5;  
    }  
    }  
    public String tostring() {  
    return "\nHourly employee: "+ super.tostring();  
    }  
     
   }
9. package Package3;  
   import Package2.\*;  
     
   public class BasePlusCommissionEmployee extends CommissionEmployee {  
    private double baseSalary;  
    BasePlusCommissionEmployee() {}  
    public BasePlusCommissionEmployee(String firstName, String lastName, String nationalIdCardNumber, double grossSales, double commissionRate, double baseSalary) {  
    super(firstName, lastName, nationalIdCardNumber, grossSales, commissionRate);  
    this.baseSalary = baseSalary;  
    }  
    public void setBaseSalary(double baseSalary){  
    this.baseSalary = baseSalary;  
    }  
    public double getBaseSalary() {  
    return baseSalary;  
    }  
    public String tostring() {  
    return "\nBase plus Commission employee: " + super.tostring();  
    }  
    public double earnings() {  
    return baseSalary + super.earnings();  
    }  
     
   }
10. package PackagePractice;  
    import Package1.\*;  
    import Package2.\*;  
    import Package3.\*;  
      
    public class PayRollSystemTest {  
     public static void main(String[] args) {  
     Employee firstEmployee = new SalariedEmployee("Muhammad", "Ali", "11111-1111", 800.00);  
     Employee secondEmployee = new CommissionEmployee("Tarwan", "Kumar",  
     "222-22-2222", 10000, 0.06);  
     Employee thirdEmployee = new BasePlusCommissionEmployee("Fabeeha", "Fatima", "333-33-3333", 5000, 0.04, 300);  
      
     Employee fourthEmployee = new HourlyEmployee("Hasnain", "Ali", "444-44-4444", 16.75, 40);  
     // polymorphism: calling tostring() and earning() on Employee’s reference  
     System.*out*.println(firstEmployee);  
     System.*out*.println(((SalariedEmployee) firstEmployee).earnings());  
     System.*out*.println(secondEmployee);  
     System.*out*.println(((CommissionEmployee) secondEmployee).earnings());  
     System.*out*.println(thirdEmployee);  
     // performing downcasting to access & raise base salary  
     BasePlusCommissionEmployee currentEmployee = (BasePlusCommissionEmployee) thirdEmployee;  
     double oldBaseSalary = currentEmployee.getBaseSalary();  
     System.*out*.println("old base salary: " + oldBaseSalary);  
     currentEmployee.setBaseSalary(1.10 \* oldBaseSalary);  
     System.*out*.println("new base salary with 10% increase is:" + currentEmployee.getBaseSalary());  
     System.*out*.println(((BasePlusCommissionEmployee) thirdEmployee).earnings());  
     System.*out*.println(fourthEmployee);  
     System.*out*.println(((HourlyEmployee) fourthEmployee).earnings());  
     } // end main  
    } // end class

Exercise 2 (Interfaces)

1. What is wrong with the following interface?

public interface SomethingIsWrong {

void aMethod(int aValue){

System.out.println("Hi Mom");

}

}

1. Fix the interface in question 1.
2. Is the following interface valid?

public interface Marker {

}

1. NO, its not because we have to make it static or default.
2. This is the correct code for firstone

public interface SomethingIsWrong {  
 default void aMethod(int aValue) {  
 System.*out*.println("Hi Mom");  
 }  
}

OR

public interface SomethingIsWrong {  
 static void aMethod(int aValue) {  
 System.*out*.println("Hi Mom");  
 }  
}

1. Yes, it is.

Exercise 3 (Interfaces)

1. Create the Animal interface.
2. Declare method legs.
3. Declare a method eat.
4. Create the Spider, Caterpillar and Cat class that implements animal interface.
5. All classes implement the Animal interface.
6. Implement the eat and legs method.
7. interface Animal {  
    abstract void legs();  
    abstract void eat();  
   }  
   class Spider implements Animal {  
    public void legs(){  
    System.*out*.println("Spider has two legs");  
    }  
    public void eat(){  
    System.*out*.println("Spider eats mosquitoes, flies, moths, and sometimes even other spiders");  
    }  
   }  
   class Caterpillar implements Animal {  
    public void legs(){  
    System.*out*.println("Caterpillar has only 6 legs");  
    }  
    public void eat(){  
    System.*out*.println("Caterpillar eats plants.");  
    }  
   }  
   class Cat implements Animal {  
    public void legs(){  
    System.*out*.println("Cat has four legs.");  
    }  
    public void eat(){  
    System.*out*.println("Cat eats Cooked beef, chicken");  
    }  
   }  
     
   public class Task3 {  
    public static void main(String[] args) {  
    Spider spider = new Spider();  
    Caterpillar caterpillar = new Caterpillar();  
    Cat cat = new Cat();  
    spider.legs();  
    spider.eat();  
    caterpillar.legs();  
    caterpillar.eat();  
    cat.legs();  
    cat.eat();  
    }  
   }

We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

abstract class Marks{  
 public abstract double getPercentage();  
}  
class A extends Marks {  
 double English;  
 double Programming;  
 double dataStructures;  
 A(double English, double Programming, double dataStructures) {  
 this.English = English;  
 this.Programming = Programming;  
 this.dataStructures = dataStructures;  
 }  
 public double getPercentage(){  
 return ((English+Programming+dataStructures)\*100/300);  
 }  
}  
class B extends Marks {  
 double English;  
 double Programming;  
 double dataStructures;  
 double Calculus;  
 B(double English, double Programming, double dataStructures, double Calculus) {  
 this.English = English;  
 this.Programming = Programming;  
 this.dataStructures = dataStructures;  
 this.Calculus = Calculus;  
 }  
 public double getPercentage(){  
 return ((English+Programming+dataStructures+Calculus)\*100/400);  
 }  
}  
public class Task4 {  
 public static void main(String[] args) {  
 A a = new A(99, 99.9, 98);  
 B b = new B(99, 99.9, 98, 92);  
 System.*out*.println("Percentage of class A. "+ a.getPercentage());  
 System.*out*.println("Percentage of class B. "+ b.getPercentage());  
 }  
}