**CSF213 Object Oriented Programming**

**Extra Practise Problems**

**Week 2 Classes And Constructors**

**Note: These problems can be done at students' own pace outside the lab hours.**

1. Create the equivalent of a four-function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation: adding, subtracting, multiplying, or dividing the two numbers. Use the concept of function overloading to have different functions calc(int,int,operator) and also calc() must be a non-static function. Finally, it should display the result. When it finishes the calculation, the program should ask if the user wants to do another calculation. The response can be Y or N.
2. 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.
3. The Java class called Holiday is started below. An object of class Holiday represents a holiday during the year. This class has three instance variables: name, day, month

a) Write a constructor for the class Holiday, which takes a String representing the name, an int representing the day, and a String representing the month as its arguments, and sets the class variables to these values.

b) Write a method inSameMonth, which compares two instances of the class Holiday, and returns the Boolean value true if they have the same month, and false if they do not.

c) Write a method avgDatewhich takes an array of base type Holiday as its argument, and returns a double that is the average of the day variables in the Holiday instances in the array. You may assume that the array is full (i.e. does not have any null entries).

d) Write a piece of code that creates a Holiday instance with the name “Independence Day”, with the day “4”, and with the month “July”.

1. The class Movie is started below. An instance of class Movie represents a film. This class has the following three class variables: title, studio, rating

a) Write a constructor for the class Movie, which takes a String representing the title of the movie, a String representing the studio, and a String representing the rating as its arguments, and sets the respective class variables to these values.

b) Write a second constructor for the class Movie, which takes a String representing the title of the movie and a String representing the studio as its arguments, and sets the respective class variables to these values, while the class variable rating is set to "PG".

c) Write a method getPG, which takes an array of base type Movie as its argument, and returns a new array of only those movies in the input array with a rating of "PG". You may assume the input array is full of Movie instances. The returned array need not be full.

d) Write a piece of code that creates an instance of the class Movie with the title “Casino Royale”, the studio “Eon Productions”, and the rating “PG13”.

1. 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.
2. 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.
3. 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. www.oumstudents.tk ◦Create an instance of Car class and initialize all the fields with appropriate values. Display the sale prices of all instance
4. Create class Number with only one private instance variable as a double primitive type. To include the following methods (include respective constructors) isZero( ), isPositive(), isNegative( ), isOdd( ), isEven( ), isPrime(), isAmstrong() the above methods return boolean primitive type. getFactorial(), getSqrt(), getSqr(), sumDigits(), getReverse() the above methods return double primitive type. void listFactor(), void dispBinary().
5. Write a constructor in the Car class given below that initializes the *brand* class field with the string “Ford”.Call the getBrand() method in the main method of the Sample class  and store the value of the brand in a variable, and print the value.
6. Create a super class called Car. The Car class has the following fields and methods. int speed; double regularPrice; String color; double getSalePrice(); 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(); // Ifweight>2000, 10%discount. Otherwise,20%discount.

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 manufacturer Discount.

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>20feet, 5%discount, Otherwise, 10%discount. Create MyOwnAutoShop class which contains the main() method. Perform the following within the main() method.