OOP Lab-05 Tasks

Name:	Syed Muhammad Raza Ali
Enrolment:	02-134231-028
Course:	OOP Lab
Faculty:	Miss Hafsa Munawar

Exercise 1 *(Computer.java)*

Write a program using the concepts of a default constructor. Consider a computer system whose name, type, processor specification, ram, hard disk drives, mother board, optical drive etc, in a constructor, desired values are entered by the user in a get method (that takes information from the user) and the displays the inputted information via display method. The user shall be asked to change any of the provided information

Code:

(ComputerSystem Class)

```
package com.mycompany.mavenproject2;

public class ComputerSystem {
    private String name,type,processor,motherBoard,opticalDrive;
    private int ram,hardDisk;

    //setters
    public void setName(String name){
        this.name = name;
    }
     public void setType(String type){
        this.type = type;
    }
      public void setProcessor(String processor){
        this.processor = processor;
    }
       public void setMotherBoard(String motherBoard){
        this.motherBoard = motherBoard;
    }
        public void setOpticalDrive(String opticalDrive){
        this.opticalDrive = opticalDrive;
    }
         public void setRAM(int ram){
        this.ram = ram;
    }
          public void sethardDisk(int hardDisk){
```

```
    this.hardDisk = hardDisk;
}

//getters
public String getName(){
    return name;
}
 public String getType(){
    return type;
}
  public String getProcessor(){
    return processor;
}
   public String getMotherBoard(){
    return motherBoard;
}
    public String getOpticalDrive(){
    return opticalDrive;
}
     public int getRAM(){
    return ram;
}
      public int getHardDisk(){
    return hardDisk;
}

 //constructor
  ComputerSystem(){
      name = "Lenovo";
      type = "Desktop";
      processor = "i7 4th gen";
      motherBoard = "MSI";
      opticalDrive = "GigaByte";
      ram = 12;
      hardDisk = 1000;
  }
  //printAll
  void printSpecs(){
      System.out.println("Name : "+name);
      System.out.println("Type : "+type);
```

```
        System.out.println("Processor : "+processor);
        System.out.println("Motherboard : "+motherBoard);
        System.out.println("OpticalDrive : "+opticalDrive);
        System.out.println("RAM : "+ram+" GB");
        System.out.println("Hard drive : "+hardDisk+" GB");
    }

}
```

Main Class

```
package com.mycompany.mavenproject2;
public class Mavenproject2 {

   public static void main(String[] args) {
      //Task---01
            ComputerSystem obj1 = new ComputerSystem();
            obj1.printSpecs();
        ComputerSystem obj = new ComputerSystem();
        obj.setName("DELL");
        obj.setType("Laptop");
        obj.setProcessor("i5 12th gen");
        obj.setOpticalDrive("HP");
        obj.setRAM(16);
        obj.sethardDisk(20000);
        System.out.println("Name : "+obj.getName());
        System.out.println("Name : "+obj.getType());

        System.out.println("Name : "+obj.getProcessor());

         System.out.println("Name : "+obj.getMotherBoard());

         System.out.println("Name : "+obj.getOpticalDrive());
          System.out.println("Name : "+obj.getRAM());
          System.out.println("Name : "+obj.getHardDisk());
)
```

Output:

By using Default Constructor

```
Name : Lenovo
Type : Desktop
Processor : i7 4th gen
Motherboard : MSI
OpticalDrive : GigaByte
RAM : 12 GB
Hard drive : 1000 GB
-------------------------
BUILD SUCCESS
```

By using Getters and Setters

```
Name : DELL
Name : Laptop
Name : i5 12th gen
Name : MSI
Name : HP
Name : 16
Name : 20000
---------------------
BUILD SUCCESS
---------------------
```

Exercise 2 Time.java)

Create a program that determines the current time and date. The program must incorporate several Methods out of which three Methods should be constructors, the first one shall be a default constructor, the second and third one shall be an overloaded constructors, from which one method deals with YEAR, MONTH AND DAY, whereas the second method deals with YEAR, MONTH,DAY,HOUR,MINUTES AND SECONDS. The other methods may include the set methods and get methods which sets and gets the described values.

Code:

Task2 Class

```
package com.mycompany.mavenproject2;

import java.time.LocalDateTime;

public class Task2 {
LocalDateTime now = LocalDateTime.now();

    private int year, day, hour, minutes, seconds;
    private String month;
    private final String[] months = {"January","February","March","April","May","June","July",
"August","September","October","November","December"};

//    Constructors
    Task2() {
        year = now.getYear();
        day = now.getDayOfMonth();
        month = now.getMonth().toString();
        month = month.substring(0, 1) + month.substring(1).toLowerCase();
        hour = now.getHour();
        minutes = now.getMinute();
        seconds = now.getSecond();
    }
        // parameterized constructor for yr, month, day
    Task2(int year, int month, int day, int hour, int minutes, int seconds) {
         this.year = year;
        this.month = months[month - 1];
        this.day = day;
        this.hour = hour;
        this.minutes = minutes;
        this.seconds = seconds;
    } // parameterized function for yr, month, day, hour, minute, second

    // Getters
    public int getYear() {
```

```
    return year;
}
public String getMonth() {
    return month;
}
public int getDay() {
    return day;
}
public int getHour() {
    return hour;
}
public int getMinutes() {
    return minutes;
}
public int getSeconds() {
    return seconds;
}

// Setters
public void setYear(int year) {
    this.year = year;
}
public void setMonth(int month) {
    this.month = months[month - 1];
}
public void setDay(int day) {
    this.day = day;
}
public void setHour(int hour) {
    this.hour = hour;
}
public void setMinute(int minutes) {
    this.minutes = minutes;
}
public void setSecond(int seconds) {
    this.seconds = seconds;
}

public void print() {
    System.out.println(month + " " + day + " " + year + " " + hour + ":" + minutes + ":" + seconds);
```

```
    }
}
```

Application Class

```
package com.mycompany.mavenproject2;
public class Mavenproject2 {
    public static void main(String[] args) {
        //Values can be taken from the user and can be validated by conditioning but giving hardcoded
        Task2 now1 = new Task2(); // default
        now1.print(); // Whole date

        Task2 now2 = new Task2(); // 3 parameterized
        now2.print(); // Whole date

        Task2 now3 = new Task2(2003, 10, 21, 11, 30, 55); // 6 parametized

//        Same setting can be done with year, hour, ,day, hour, minutes, seconds
        String month = now3.getMonth(); // getting
        System.out.println("The current month is " + month);
        Scanner input = new Scanner(System.in);
        System.out.print("Please enter a number that indicates month: "); // setting
        int monthInput = input.nextInt();
        now3.setMonth(monthInput);
        month = now3.getMonth(); // getting again
        System.out.println("Now the month is " + month);

        System.out.println("And the whole date looks like:");
        now3.print(); // Whole date
}
```

Output:

```
October 21 2023 7:24:58
October 21 2023 7:24:58
The current month is October
Please enter a number that indicates month: 9
Now the month is September
And the whole date looks like:
September 21 2003 11:30:55
------------------------------------------------
BUILD SUCCESS
```

Exercise 3 (Book.java)

Write a Java class Book with following features:

- Instance variables :
  - **title** for the title of book of type String.
  - **author** for the author's name of type String.
  - **price** for the book price of type double.
- Constructor:
  - **public Book (String title, Author name, double price)**: A constructor with parameters, it creates the Author object by setting the the fields to the passed values.
- Instance methods:
  - **public void setTitle(String title)**: Used to set the title of book.
  - **public void setAuthor(String author)**: Used to set the name of author of book.
  - **public void setPrice(double price)**: Used to set the price of book.
  - **public double getTitle()**: This method returns the title of book.
  - **public double getAuthor()**: This method returns the author's name of book.
  - **public String toString()**: This method printed out book's details to the screen

Write a separate class **BookDemo** with a main() method creates a Book titled "Developing Java Software" with authors Russel Winderand price 79.75. Prints the Book's string representation to standard output (using System.out.println).

Code:

Book Class

```
package com.mycompany.mavenproject2;

class Book{
   private String title, author;
   private double price;

   //setters
   public void setTitle(String title){
      this.title = title;
   }
   public void setAuthor(String author){
      this.author = author;
   }
   public void setPrice(double price){
```

```
        this.price = price;
    }

    //getters
    public String getTitle(){
        return title;
    }
        public String getAuthor(){
        return author;
    }
            public double getPrice(){
        return price;
    }
            //constructor
            Book(){
                title = "Developing Java Software";
                author = "Russel Winderand";
                price = 79.75;
            }

    @Override
            public String toString(){
                return ("Name : "+title+"\n"+"Author : "+author+"\n"+"Price : "+price);
            }
}
```

Application Class

```
package com.mycompany.mavenproject2;
public class Mavenproject2 {
    public static void main(String[] args) {
        Book book1 = new Book();
        System.out.println("=============== By using Getters and Setters ===============");
        book1.setTitle("Fundamentals of Physics");
        book1.setAuthor("Halliday Resnick");
        book1.setPrice(100.32);
        System.out.println("Title : "+book1.getTitle());
        System.out.println("Title : "+book1.getAuthor());
        System.out.println("Title : "+book1.getPrice());

        System.out.println("=============== By using Default constructor ===============");
        Book book2 = new Book();
        System.out.println(book2);
    }
}
```

Output:

```
=============== By using Getters and Setters ===============
Title : Fundamentals of Physics
Title : Halliday Resnick
Title : 100.32
=============== By using Default constructor ===============
Name : Developing Java Software
Author : Russel Winderand
Price : 79.75
------------------------------------------------------------------------
BUILD SUCCESS
```