|  |
| --- |
| **Lab04: Classes and Objects** |

Designing and implementing Java programs that deal with:

|  |
| --- |
| 1. A class Definition 2. Constructors 3. Class members 4. Class methods 5. The new Operator to create a new object 6. The this Operator |

|  |
| --- |
| **Exercises** |

Exercise 1 (Computer.java)

Write a program using the concepts of a default constructor. Consider a computer system's 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 then displays the inputted information via display method. The user shall be asked to change any of the provided information.

Source Code:

package computersystemmain;

import java.util.Scanner;

class ComputerSystem {

String name;

String type;

String processor;

String ram;

String hdd;

String motherboard;

String opticalDrive;

public ComputerSystem() {

this.name = "Default Name";

this.type = "Desktop";

this.processor = "Intel Core i5";

this.ram = "8GB";

this.hdd = "1TB";

this.motherboard = "ASUS";

this.opticalDrive = "DVD RW";

}

public void getDetails() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter Computer Name: ");

this.name = scanner.nextLine();

System.out.print("Enter Computer Type (Desktop/Laptop): ");

this.type = scanner.nextLine();

System.out.print("Enter Processor Specification: ");

this.processor = scanner.nextLine();

System.out.print("Enter RAM size: ");

this.ram = scanner.nextLine();

System.out.print("Enter Hard Disk Drive (HDD) size: ");

this.hdd = scanner.nextLine();

System.out.print("Enter Motherboard: ");

this.motherboard = scanner.nextLine();

System.out.print("Enter Optical Drive: ");

this.opticalDrive = scanner.nextLine();

}

public void displayDetails() {

System.out.println("\nComputer System Details:");

System.out.println("Name: " + this.name);

System.out.println("Type: " + this.type);

System.out.println("Processor: " + this.processor);

System.out.println("RAM: " + this.ram);

System.out.println("HDD: " + this.hdd);

System.out.println("Motherboard: " + this.motherboard);

System.out.println("Optical Drive: " + this.opticalDrive);

}

public void changeDetails() {

Scanner scanner = new Scanner(System.in);

String choice;

System.out.println("\nWould you like to change any details? (yes/no)");

choice = scanner.nextLine();

if (choice.equalsIgnoreCase("yes")) {

System.out.println("Which detail would you like to change?");

System.out.println("1. Name");

System.out.println("2. Type");

System.out.println("3. Processor");

System.out.println("4. RAM");

System.out.println("5. HDD");

System.out.println("6. Motherboard");

System.out.println("7. Optical Drive");

System.out.print("Enter the number corresponding to the detail you want to change: ");

int option = scanner.nextInt();

scanner.nextLine();

switch (option) {

case 1:

System.out.print("Enter new Computer Name: ");

this.name = scanner.nextLine();

break;

case 2:

System.out.print("Enter new Computer Type (Desktop/Laptop): ");

this.type = scanner.nextLine();

break;

case 3:

System.out.print("Enter new Processor Specification: ");

this.processor = scanner.nextLine();

break;

case 4:

System.out.print("Enter new RAM size: ");

this.ram = scanner.nextLine();

break;

case 5:

System.out.print("Enter new HDD size: ");

this.hdd = scanner.nextLine();

break;

case 6:

System.out.print("Enter new Motherboard: ");

this.motherboard = scanner.nextLine();

break;

case 7:

System.out.print("Enter new Optical Drive: ");

this.opticalDrive = scanner.nextLine();

break;

default:

System.out.println("Invalid option.");

}

}

}

}

public class ComputerSystemMain {

public static void main(String[] args) {

ComputerSystem computer = new ComputerSystem();

computer.getDetails();

computer.displayDetails();

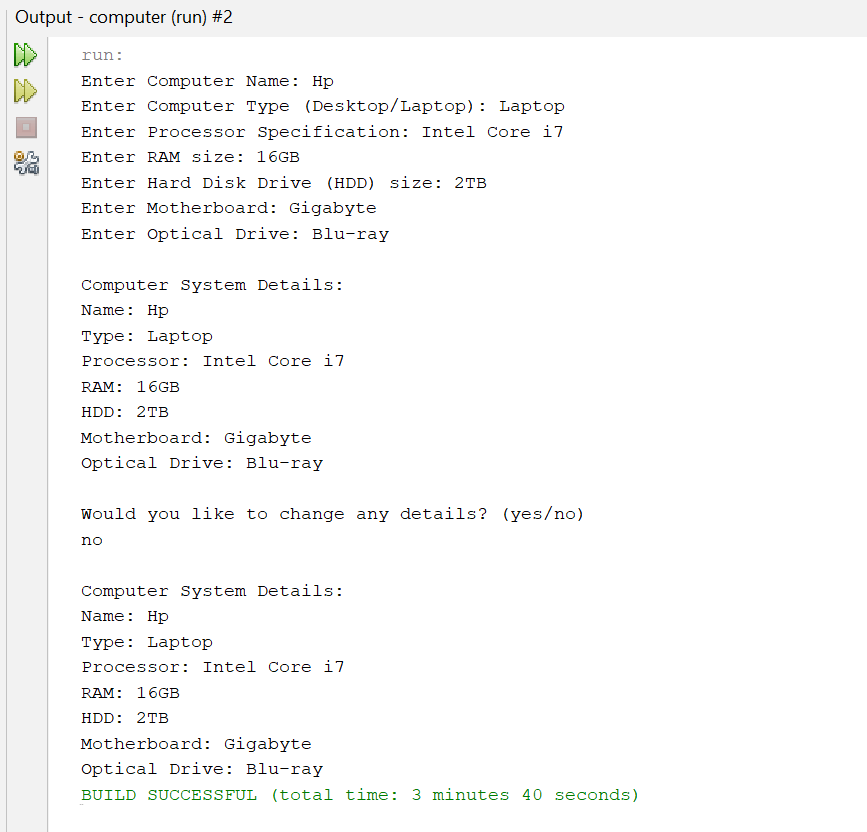
computer.changeDetails();

computer.displayDetails();

}

}

Output:



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.

Source Code:

package time;

import java.time.LocalDateTime;

class CurrentDateTime {

private int year;

private int month;

private int day;

private int hour;

private int minute;

private int second;

public CurrentDateTime() {

LocalDateTime now = LocalDateTime.now();

this.year = now.getYear();

this.month = now.getMonthValue();

this.day = now.getDayOfMonth();

this.hour = now.getHour();

this.minute = now.getMinute();

this.second = now.getSecond();

}

public CurrentDateTime(int year, int month, int day) {

this.year = year;

this.month = month;

this.day = day;

LocalDateTime now = LocalDateTime.now(); // Use current time for hours, minutes, and seconds

this.hour = now.getHour();

this.minute = now.getMinute();

this.second = now.getSecond();

}

public CurrentDateTime(int year, int month, int day, int hour, int minute, int second) {

this.year = year;

this.month = month;

this.day = day;

this.hour = hour;

this.minute = minute;

this.second = second;

}

public int getYear() {

return year;

}

public int getMonth() {

return month;

}

public int getDay() {

return day;

}

public int getHour() {

return hour;

}

public int getMinute() {

return minute;

}

public int getSecond() {

return second;

}

public void setYear(int year) {

this.year = year;

}

public void setMonth(int month) {

this.month = month;

}

public void setDay(int day) {

this.day = day;

}

public void setHour(int hour) {

this.hour = hour;

}

public void setMinute(int minute) {

this.minute = minute;

}

public void setSecond(int second) {

this.second = second;

}

public void displayDateTime() {

System.out.println("Current Date and Time: " + year + "-" + month + "-" + day + " " + hour + ":" + minute + ":" + second);

}

}

public class TimeMain {

public static void main(String[] args) {

CurrentDateTime defaultDateTime = new CurrentDateTime();

System.out.println("Default Constructor:");

defaultDateTime.displayDateTime();

CurrentDateTime customDateTime1 = new CurrentDateTime(2024, 10, 18);

System.out.println("\nOverloaded Constructor (YEAR, MONTH, DAY):");

customDateTime1.displayDateTime();

CurrentDateTime customDateTime2 = new CurrentDateTime(2024, 10, 18, 10, 45, 30);

System.out.println("\nOverloaded Constructor (YEAR, MONTH, DAY, HOUR, MINUTE, SECOND):");

customDateTime2.displayDateTime();

System.out.println("\nSetting new date and time values:");

customDateTime2.setYear(2025);

customDateTime2.setMonth(5);

customDateTime2.setDay(20);

customDateTime2.setHour(12);

customDateTime2.setMinute(30);

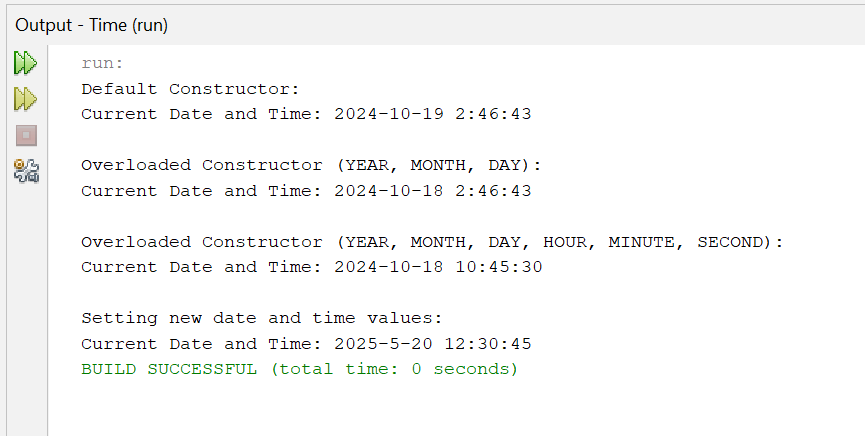
customDateTime2.setSecond(45);

customDateTime2.displayDateTime();

}

}

Output:



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 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).

SOURCE CODE:

package book;

class Book {

// Instance variables

private String title;

private String author;

private double price;

// Constructor

public Book(String title, String author, double price) {

this.title = title;

this.author = author;

this.price = 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;

}

public String toString() {

return "Book Title: " + title + "\nAuthor: " + author + "\nPrice: $" + price;

}

}

// BookDemo Class

public class BookDemo {

public static void main(String[] args) {

// Creating a book object using the constructor

Book myBook = new Book("Developing Java Software", "Russel Winder", 79.75);

// Printing the book's details using the toString method

System.out.println(myBook.toString());

}

}

OUTPUT:

