

**Faculty of Engineering and Technology**  
**University of Sindh**

**Department of :** Data Science

**Class:** 2k24/106/DS **Part 1 (Morning)**

**Semester:** Second Semester

**Course:** Object-Oriented Programming

**Full Name:** Mohammad Hassan Qazi

**Course instructor :** Nazish bashir

**Roll No.:** 106

**Submission Date :** 24/12/2k24

---

**Course Instructor:** Nazish **Department of Information Technology, University of Sindh.**

## ASSIGNMENT AND LAB EXERCISES OBJECT ORIENTED PROGRAMMING ( 320 / 321 )

### WEEK 1 HANDOUT EXERCISE:

**EXERCISE 1-1:** Write a program which displays information about you like: Your Name, Fathers Name, Class Roll No., age, and Cell Phone number. Use only println() method.

//CODE

```
import java.util.Scanner;

public class Example {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println("Enter first number: ");

        int num1 = input.nextInt();

        System.out.println("Enter second number: ");

        int num2 = input.nextInt();

        int sum = num1 + num2;

        System.out.println("Sum: " + sum);

        input.close();

    }

}
```

**EXERCISE 1-2:** Write down a program which has 2 variable 'a' and 'b' assign them any value and evaluate the following equation:  $x = a^2 + 2ab + b^2$  Create variable x and Display the result of 'x'

public class Exercise1\_2

//CODE

```
public class Exercise1_2 {

    public static void main(String[] args) {

        int a = 3;

        int b = 4;

        int x = (a * a) + (2 * a * b) + (b * b);

        System.out.print("Result of x: " + x);  } }
```

## WEEK 2 HANDOUT EXERCISE:

**EXERCISE 2-1: 1.** Declare two short variable and try to assign their sum to byte variable and write down what happened.

//Code

```
short a=10, b=20;
```

```
byte c = a+b;
```

```
byte result = (byte) (a + b);
```

will cause a compile time error

**2.** Declare a long variable with value with int range then declare another long variable with value outside the range of int. Write down what happened.

//CODE

```
public class LongVariableExample {
```

```
    public static void main(String[] args) {
```

```
        long withinIntRange = 2147483647; // Maximum value for int
```

```
        long outsideIntRange = 2147483648L; // Value exceeding int range
```

```
        System.out.print("Within int range: " + withinIntRange);
```

```
        System.out.print("\nOutside int range: " + outsideIntRange);
```

```
    }
```

```
}
```

**Within int range:** The value is assigned successfully as it fits within the int range.

**Outside int range:** Requires **L** to specify it as a long literal; without it, a **compiler error** occurs due to exceeding the int range.

**3.** Declare a float variable like this and write down what happened. float a = 12.54;

**Output**

```
float a = 12.54f;
```

**Runs successfully**

**4.** Declare all types of integers and floating point variables and try to give those values outside of their range and then write down the errors

```
public class DataTypeRangeTest {
```

```
    public static void main(String[] args) {
```

```
        byte b = 128;
```

```

short s = 32768;

int i = 2147483648;
long l = 9223372036854775808L;

// Floating-Point Types

float f = 3.4028236E39f;

double d = 1.7976931348623159E309;

System.out.println(b);

System.out.println(s);

System.out.println(i);

System.out.println(l);

System.out.println(f);

System.out.println(d);
}
}

```

#### Errors Observed:

- byte b = 128; - Error: Value out of range for byte (-128 to 127).
- short s = 32768; - Error: Value out of range for short (-32,768 to 32,767).
- int i = 2147483648; - Error: Integer number too large for int.
- long l = 9223372036854775808L; - Error: Integer number too large for long.
- float f = 3.4028236E39f; - Error: Value exceeds the float range.
- double d = 1.7976931348623159E309; - Error: Value exceeds the double range.

5. Create 5 character variable by giving it different hexadecimal codes and write down what characters are shown on screen.

//Code

```

public class HexadecimalCharExample {

    public static void main(String[] args) {

        char char1 = '\u0041'; // Hexadecimal for 'A'

        char char2 = '\u0042'; // Hexadecimal for 'B'

        char char3 = '\u0043'; // Hexadecimal for 'C'

        char char4 = '\u0044'; // Hexadecimal for 'D'
    }
}

```

```

char char5 = '\u0045'; // Hexadecimal for 'E'

System.out.print(char1);

System.out.print(char2);

System.out.print(char3);

System.out.print(char4);

System.out.print(char5);

}

```

**output**

ABCDE

**6.** Declare and initialize two character variable with any alphabet value. Then use this statement to print the result and write down what happened. `System.out.println (ch1 + ch2);`

```

//code

public class Main {

    public static void main(String[] args) {

        char ch1 = 'A';

        char ch2 = 'B';

        System.out.println(ch1 + ch2); // Output will be an integer

    }

}

```

**What happens:**

- Since ch1 and ch2 are characters, Java automatically converts them to their corresponding integer Unicode values before performing the addition.
- The result is an integer, not a character, and is printed as such.

```

int c = 15; // Example value for c
int d = 4; // Example value for d

float ans = (float) a / b + (float) c / d;

System.out.println("Result of a / b + c / d = " + ans);

}

}

```

**EXERCISE 2-2:** Write down a program which has 4 int variables a, b, c and d with different values. Then store the result of equation “a\ b + c\ d” in a variable “ans” of float data type. The result should be accurate.

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int a = 10; // Example value for a  
  
        int b = 3; // Example value for b  
  
        int c = 15; // Example value for c  
  
        int d = 4; // Example value for d  
  
        float ans = (float) a / b + (float) c / d;  
  
        System.out.println("Result of a / b + c / d = " + ans);  
  
    }  
}  
  
//output  
Result of a / b + c / d = 7.0833335
```

### **WEEK 3 HANDOUT EXERCISE:**

**EXERCISE 3-1:** Write down a program which has an array of 10 integer numbers then write the code to find the largest and smallest number from that array

```
public class Main {  
  
    public static void main(String[] args) {  
  
        int[] numbers = {12, 45, 7, 89, 34, 23, 67, 98, 56, 31};  
  
        int largest = numbers[0], smallest = numbers[0];  
  
        for (int num : numbers) {  
  
            largest = Math.max(largest, num);  
  
            smallest = Math.min(smallest, num);  
  
        }  
  
        System.out.println("Largest: " + largest + "\nSmallest: " + smallest);  
  
    }  
}
```

**WEEK 4 HANDOUT EXERCISE:**

**EXERSICE 4-1:** Write down a program which takes the temperature in Celsius °C from the user and displays the output in Fahrenheit °F °F = (°C × 9/5) + 32

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter temperature in Celsius: ");

        double celsius = scanner.nextDouble();

        double fahrenheit = (celsius * 9/5) + 32;

        System.out.println("Temperature in Fahrenheit: " + fahrenheit)

        scanner.close();

    }

}
```

**WEEK 6 HANDOUT EXERCISE:****EXERSICE 6-1:**

Create a program for Mark sheet with following requirements: • Take input for data of 5 students in 7 different arrays like: (Roll Number, Name, Marks of 5 different subjects

**import java.util.Scanner;**

import java.util.Scanner;

```
public class MarkSheet {

    public static void main(String[] args) {

        // Arrays to store student data

        String[] rollNumbers = new String[5];

        String[] names = new String[5];

        int[] marksEnglish = new int[5];

        int[] marksCalculus = new int[5];

        int[] marksElectronics = new int[5];

        int[] marksProgramming = new int[5];

    }

}
```

```
int[] marksIntroIT = new int[5];

// Create a Scanner object to take input from the user
Scanner scanner = new Scanner(System.in);

// Input data for 5 students
for (int i = 0; i < 5; i++) {

    System.out.println("Enter details for Student " + (i + 1) + ":");

    // Input roll number and name
    System.out.print("Enter Roll Number: ");
    rollNumbers[i] = scanner.nextLine();

    System.out.print("Enter Name: ");
    names[i] = scanner.nextLine();

    // Input marks for 5 subjects
    System.out.print("Enter marks for English: ");
    marksEnglish[i] = scanner.nextInt();

    System.out.print("Enter marks for Calculus: ");
    marksCalculus[i] = scanner.nextInt();

    System.out.print("Enter marks for Electronics: ");
    marksElectronics[i] = scanner.nextInt();

    System.out.print("Enter marks for Programming: ");
    marksProgramming[i] = scanner.nextInt();

    System.out.print("Enter marks for Intro. to IT: ");
    marksIntroIT[i] = scanner.nextInt();
```



```

        // Move to the next line after reading integers
        scanner.nextLine();
    }

    // Close the scanner
    scanner.close();

    // Display entered data (optional)
    for (int i = 0; i < 5; i++) {
        System.out.println("\nDetails for Student " + (i + 1) + ":");
        System.out.println("Roll Number: " + rollNumbers[i]);
        System.out.println("Name: " + names[i]);
        System.out.println("Marks in English: " + marksEnglish[i]);
        System.out.println("Marks in Calculus: " + marksCalculus[i]);
        System.out.println("Marks in Electronics: " + marksElectronics[i]);
        System.out.println("Marks in Programming: " + marksProgramming[i]);
        System.out.println("Marks in Intro. to IT: " + marksIntroIT[i]);
    }
}

```

**} 2 : Calculate and store the Total, Percentage and Grade of them in other 3 arrays.**

```

public class MarkSheet {

    public static void main(String[] args) {

        // Arrays to store student data
        int[] marksEnglish = {80, 85, 70, 90, 88};
        int[] marksCalculus = {70, 88, 75, 95, 80};
        int[] marksElectronics = {90, 91, 85, 92, 86};
        int[] marksProgramming = {85, 90, 80, 93, 89};
        int[] marksIntroIT = {75, 80, 78, 85, 91};
        int[] totals = new int[5];
        double[] percentages = new double[5];
    }
}

```

```
char[] grades = new char[5];

// Loop through each student (5 students)
for (int i = 0; i < 5; i++) {

    // Calculate total for each student
    totals[i] = marksEnglish[i] + marksCalculus[i] + marksElectronics[i] + marksProgramming[i] + marksIntroIT[i];
    percentages[i] = (double) totals[i] / 5;

    // Assign grade based on percentage
    if (percentages[i] >= 90) {
        grades[i] = 'A';
    } else if (percentages[i] >= 80) {
        grades[i] = 'B';
    } else if (percentages[i] >= 70) {
        grades[i] = 'C';
    } else if (percentages[i] >= 60) {
        grades[i] = 'D';
    } else {
        grades[i] = 'F';
    }
}

}}
```

**3** : Use loops, and decision statements for the repetition and decision making.

```
public class MarkSheet {  
    public static void main(String[] args) {  
        // Arrays for storing marks for 5 students  
        int[] marksEnglish = {80, 85, 70, 90, 88};  
        int[] marksCalculus = {70, 88, 75, 95, 80};  
        int[] marksElectronics = {90, 91, 85, 92, 86};  
        int[] marksProgramming = {85, 90, 80, 93, 89};  
        int[] marksIntrolT = {75, 80, 78, 85, 91};  
        int[] totals = new int[5];  
        double[] percentages = new double[5];  
        char[] grades = new char[5];  
  
        for (int i = 0; i < 5; i++) {  
            totals[i] = marksEnglish[i] + marksCalculus[i] + marksElectronics[i] + marksProgramming[i] + marksIntrolT[i];  
            percentages[i] = (double) totals[i] / 5;  
            if (percentages[i] >= 90) {  
                grades[i] = 'A';  
            } else if (percentages[i] >= 80) {  
                grades[i] = 'B';  
            } else if (percentages[i] >= 70) {  
                grades[i] = 'C';  
            } else if (percentages[i] >= 60) {  
                grades[i] = 'D';  
            } else {  
                grades[i] = 'F';  
            }  
        }  
    }  
}
```

Then display the results as shown below

Enter details for student 1

Enter Roll Number: 2k24/DS/106

Enter Name of Student: Hassan Qazi

Enter Marks in English: 80

Enter Marks in Calculus: 70

Enter Marks in Electronics: 90

Enter Marks in Programming: 85

Enter Marks in Intro. to IT: 75

Total Marks: 400

Percentage: 80.00%

Grade: B

**EXERCISE 6-2:** Write a program to input a character from user and check whether given character alphabet, digit or special character is using if else. If the user presses “ @ ” the program should exit. Input character: a 'a' is alphabet Input character: A 'A' is alphabet Input character: # '#' is special character Input character: 1 '1' is a number Input character: @ Exit

**//CODE**

```
import java.util.Scanner;

public class CharacterChecker {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        while (true) {

            System.out.print("Input character: ");

            char input = scanner.next().charAt(0);

            if (input == '@') {

                System.out.println("Exit");

                break; // Exit the loop and terminate the program

            }

            if ((input >= 'a' && input <= 'z') || (input >= 'A' && input <= 'Z')) {
```

```

        System.out.println("'" + input + "' is an alphabet");
    }

    else if (input >= '0' && input <= '9') {
        System.out.println("'" + input + "' is a number");
    }

    else {
        System.out.println("'" + input + "' is a special character");
    }
}

scanner.close();
}
}

```

#### WEEK 7 HANDOUT EXERCISE:

**Exercise 7-1: EXERCISE 7-1: Write a class encapsulating the concept of a course grade, assuming a course grade has the following attributes:**

- **course name**
- **letter grade**

**. Include a constructor, and include the following methods:**

- **A method to take input in course name and grade (restrict user to give wrong values of grade correct grade value will be: "A, B, C, D and F")**
- **A method to display course name and grade.**

```

import java.util.Scanner;

public class CourseGrade
{
    private String courseName;
    private char letterGrade;

    public CourseGrade()
    {

```

```
{  
    courseName = "";  
    letterGrade = 'F'; // Default grade  
}  
  
public void inputDetails() {  
    Scanner scanner = new Scanner(System.in);  
    System.out.print("Enter course name: ");  
    courseName = scanner.nextLine();  
    while (true) {  
        System.out.print("Enter grade (A, B, C, D, F): ");  
        char grade = scanner.next().charAt(0);  
        grade = Character.toUpperCase(grade); // Convert to uppercase for consistency  
        if (grade == 'A' || grade == 'B' || grade == 'C' || grade == 'D' || grade == 'F') {  
            letterGrade = grade;  
            break; // Exit loop if input is valid  
        } else {  
            System.out.println("Invalid grade! Please enter a valid grade (A, B, C, D, F).");  
        }  
    }  
}  
  
// Method to display course details  
public void displayDetails() {  
    System.out.println("Course Name: " + courseName);  
    System.out.println("Letter Grade: " + letterGrade);  
}  
}
```

**Finally write a client class to test all the methods in your class.**

```
import java.util.Scanner;

public class CourseGrade {

    private String courseName;

    private char letterGrade;

    public CourseGrade() {

        courseName = ""; // Default course name

        letterGrade = 'F'; // Default grade

    }

    public void inputDetails() {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter course name: ");

        courseName = scanner.nextLine();

        while (true) {

            System.out.print("Enter grade (A, B, C, D, F): ");

            char grade = scanner.next().charAt(0);

            grade = Character.toUpperCase(grade); // Convert to uppercase for consistency

            if (grade == 'A' || grade == 'B' || grade == 'C' || grade == 'D' || grade == 'F') {

                letterGrade = grade;

                break; // Exit loop if input is valid

            } else {

                System.out.println("Invalid grade! Please enter a valid grade (A, B, C, D, F).");

            }

        }

    }

}
```

```

public void displayDetails() {
    System.out.println("Course Name: " + courseName);
    System.out.println("Letter Grade: " + letterGrade);
}
}

```

## WEEK 8 EXTRA EXERCISE:

Explore more methods of String class by searching on internet and do the following tasks:

- Task 1: String Replacement** 1. Create a String object with the value "Hello, World!".
2. Use the `replace()` method to replace all occurrences of "World" with "Universe".
3. Use the `replaceAll()` method to replace all occurrences of "l" with "x".
4. Display the modified strings.

*//code*

```

public class StringReplacement {
    public static void main(String[] args) {
        String str = "Hello, World!";
        String replacedStr1 = str.replace("World", "Universe");
        System.out.println("After replace(): " + replacedStr1);
        String replacedStr2 = str.replaceAll("l", "x");
        System.out.println("After replaceAll(): " + replacedStr2);
    }
}

```

## Task 2: String Splitting

1. Create a String object with the value "apple,banana,cherry,orange".
2. Use the `split()` method to split the string into an array of substrings using the comma as a delimiter.
3. Use the `split()` method with a regular expression to split the string into an array of substrings.
4. Display the contents of the arrays.



```

public class StringSplitting {

    public static void main(String[] args) {

        // Original string

        String str = "apple,banana,cherry,orange";


        // Splitting using comma as a delimiter

        String[] fruits1 = str.split(","); // Using ',' directly

        System.out.println("Split using comma:");

        for (String fruit : fruits1) {

            System.out.println(fruit);

        }


        // Splitting using a regular expression (same result here)

        String[] fruits2 = str.split("\\s*,\\s*"); // Handles spaces around commas

        System.out.println("\nSplit using regex:");

        for (String fruit : fruits2) {

            System.out.println(fruit);

        }

    }

}

```

**Task 3: String Joining 1. Create an array of strings containing the values "apple", "banana", "cherry".**

**2. Use the String.join() method to join the array of strings into a single string using the comma as a delimiter.**

**3. Display the joined string.**

```

public class StringJoining {

    public static void main(String[] args) {

        String[] fruits = {"apple", "banana", "cherry"};

        String joinedString = String.join(",", fruits);

        System.out.println("Joined String: " + joinedString);    } }

```

#### Task 4:

##### String Formatting

1. Create a String object with the value "Hello, %s!".
2. Use the String.format() method to replace the placeholder with the value "World".
3. Display the formatted string.

```
public class StringFormatting {  
    public static void main(String[] args) {  
        String str = "Hello, %s!";  
        String formattedString = String.format(str, "World");  
        System.out.println("Formatted String: " + formattedString);  
    }  
}
```

#### WEEK 11 EXTRA EXERCISE:

Write the code for following class diagram. Class names, variables names and method names should be same as given in diagram. You may add proper logical code of your choice in each method.

**//code for person**

```
public class Person {  
    String name;  
    int age;  
    String gender;  
    public Person(String name, int age, String gender) {  
        this.name = name;  
        this.age = age;  
        this.gender = gender;  
    }  
    public void printData() {  
        System.out.println("Name: " + name);  
        System.out.println("Age: " + age);  
        System.out.println("Gender: " + gender);  
    }  
}
```

```

    }

    public void login() {
        System.out.println(name + " logged in successfully!");
    }
}

//Code for teacher

public class Teacher extends Person {
    int empno;
    double salary;

    public Teacher(String name, int age, String gender, int empno, double salary) {
        super(name, age, gender);
        this.empno = empno;
        this.salary = salary;
    }

    public void printPaySlip() {
        System.out.println("Employee Number: " + empno);
        System.out.println("Salary: " + salary);
    }

    @Override
    public void printData() {
        super.printData();
        System.out.println("Employee Number: " + empno);
    }
}

```

//Code for student

```
public class Student extends Person {  
    int rollno;  
    String grade;  
    public Student(String name, int age, String gender, int rollno, String grade) {  
        super(name, age, gender); // Calling Person class constructor  
        this.rollno = rollno;  
        this.grade = grade;  
    }  
    public void showResult() {  
        System.out.println("Roll Number: " + rollno);  
        System.out.println("Grade: " + grade);  
    }  
    @Override  
    public void printData() {  
        super.printData();  
        System.out.println("Roll Number: " + rollno);  
    }  
}
```

## THEORY ASSIGNMENT (10 MARKS):

### Task 1: Displaying the Current Date and Time

1. Import the necessary classes: `java.util.Date` and `java.text.SimpleDateFormat`.
2. Create a `Date` object to represent the current date and time.
3. Create a `SimpleDateFormat` object to format the date and time.
4. Use the `format()` method to format the date and time according to the specified pattern.
5. Display the formatted date and time.

```
import java.util.Date;

import java.text.SimpleDateFormat;

public class DisplayDateTime {

    public static void main(String[] args) {

        Date currentDate = new Date();

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

        String formattedDate = dateFormat.format(currentDate);

        System.out.println("Current Date and Time: " + formattedDate);

    }
}
```

### Task 2: Formatting Dates and Times

1. Create a `Date` object to represent a specific date and time (e.g., January 1, 2024, 12:00:00 PM).
2. Create multiple `Simple Date Format` objects to format the date and time according to different patterns (e.g., "yyyy-MM-dd", "MM/dd/yyyy", "EEEE, MMMM dd, yyyy").
3. Use the `format()` method to format the date and time according to each pattern.
4. Display the formatted dates and times.

```
import java.util.Date;

import java.text.SimpleDateFormat;

public class FormattingDates {

    public static void main(String[] args) {

        Date specificDate = new Date(124, 0, 1, 12, 0, 0); // Year: 2024 - 1900, Month: 0 (January), Day: 1, Hour: 12,
        Minute: 0, Second: 0

        SimpleDateFormat dateFormat1 = new SimpleDateFormat("yyyy-MM-dd");

        SimpleDateFormat dateFormat2 = new SimpleDateFormat("MM/dd/yyyy");
    }
}
```

```

SimpleDateFormat dateFormat3 = new SimpleDateFormat("EEEE, MMMM dd, yyyy");

String formattedDate1 = dateFormat1.format(specificDate);

String formattedDate2 = dateFormat2.format(specificDate);

String formattedDate3 = dateFormat3.format(specificDate);

System.out.println("Formatted Date (yyyy-MM-dd): " + formattedDate1);

System.out.println("Formatted Date (MM/dd/yyyy): " + formattedDate2);

System.out.println("Formatted Date (EEEE, MMMM dd, yyyy): " + formattedDate3);

}

}

```

### Task 3: Parsing Dates and Times

1. Create a `SimpleDateFormat` object to parse a date and time string according to a specific pattern (e.g., "yyyy-MM-dd").
2. Create a `Date` object by parsing a date and time string using the `parse()` method.
3. Display the parsed date and time.

```

import java.text.SimpleDateFormat;

import java.util.Date;

public class ParsingDates {

    public static void main(String[] args) {

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd");

        try {

            String dateString = "2024-01-01"; // Date string in the format "yyyy-MM-dd"

            Date parsedDate = dateFormat.parse(dateString);

            System.out.println("Parsed Date: " + parsedDate);

        } catch (Exception e) {

            System.out.println("Error parsing date: " + e.getMessage());

        }

    }

}

```

#### Task 4: Calculating Date and Time Intervals

1. Create two Date objects to represent two different dates and times.
2. Calculate the interval between the two dates and times using the `getTime()` method.
3. Display the interval in a human-readable format (e.g., "X days, Y hours, Z minutes, W seconds").

```
import java.util.Date;

import java.util.concurrent.TimeUnit;

public class DateTimeInterval {

    public static void main(String[] args) {

        Date date1 = new Date(124, 0, 1, 12, 0, 0); // January 1, 2024, 12:00:00 PM
        Date date2 = new Date(124, 0, 2, 14, 30, 0); // January 2, 2024, 2:30:00 PM

        long timeDifference = date2.getTime() - date1.getTime(); // Difference in milliseconds

        long diffInSeconds = TimeUnit.MILLISECONDS.toSeconds(timeDifference);
        long diffInMinutes = TimeUnit.MILLISECONDS.toMinutes(timeDifference);
        long diffInHours = TimeUnit.MILLISECONDS.toHours(timeDifference);
        long diffInDays = TimeUnit.MILLISECONDS.toDays(timeDifference);

        long remainingHours = diffInHours - diffInDays * 24;

        long remainingMinutes = diffInMinutes - TimeUnit.HOURS.toMinutes(diffInHours);
        long remainingSeconds = diffInSeconds - TimeUnit.MINUTES.toSeconds(diffInMinutes);

        System.out.println("Interval: " + diffInDays + " days, " + remainingHours + " hours, "
            + remainingMinutes + " minutes, " + remainingSeconds + " seconds.");
    }
}
```