**NAME: SOWJANYA BS**

**DAY 2 - ASSIGNMENT**

**SECTION 1 (Decision Making)**

**1.**  **Problem Statement** : Everyday Decisions

Think of three common decisions you make daily.

Write them in the format: If [condition], then [action].

How would a computer represent these decisions using Java syntax?

**Pseudo code** **:**

Start

create scanner to get user input

ask "are you hungry?"

read input as hungry

if hungry is "yes" then

print "you should eat"

else

print "you don't need to eat right now"

ask "are you tired?"

read input as tired

if tired is "yes" then

print "you should take rest or sleep"

else

print "you have enough energy"

ask "do you have homework?"

read input as homework

if homework is "yes" then

print "you should study now"

else

print "you can relax or do something fun"

close the scanner

End

**Algorithm: steps**

1. Start the program.
2. Create a Scanner object for user input.
3. Ask the user: "Are you hungry?" Read input.
4. If input is yes, print "You should eat", else print "You don't need to eat right now."
5. Ask the user: Are you tired? Read input.
6. If input is yes, print "You should take rest or sleep", else print "You have enough energy."
7. Ask the user: "Do you have homework?" Read input.
8. If input is yes, print "You should study now", else print "You can relax or do something fun."
9. Close the Scanner.
10. End the program.

**Code** :

import java.util.Scanner;

public class EverydayDecisions {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Are you hungry? (yes/no): ");

String hungry = scanner.nextLine().toLowerCase();

if (hungry.equals("yes")) {

System.out.println("You should eat.");

} else {

System.out.println("You don't need to eat right now.");

}

System.out.print("Are you tired? (yes/no): ");

S String tired = scanner.nextLine().toLowerCase();

if (tired.equals("yes")) {

System.out.println("You should take rest or sleep.");

} else {

System.out.println("You have enough energy.");

}

System.out.print("Do you have homework? (yes/no): ");

String homework = scanner.nextLine().toLowerCase();

if (homework.equals("yes")) {

System.out.println("You should study now.");

} else {

System.out.println("You can relax or do something fun.");

}

scanner.close();

}

}

| Test cases | Input / Expected Output |
| --- | --- |
| TC1 |  |
| TC2 |  |
| TC3 |  |

**Observation**

* The program uses simple if conditions to simulate daily decisions like eating, resting, and studying based on user input.
* It clearly shows how real-life choices can be represented in code, making it easy to understand and extend.

**2.**  **Problem Statement** :Imagine a simple smart home device.

Write two decisions it might need to make in daily use.

07

09

3

Describe the condition and the corresponding action in pseudocode or Java code.

**Pseudo code** **:**

Start

create scanner to read user input

prompt user to enter current time

read time input

prompt user to enter current temperature

read temperature input

if time equals 19 then

turn on lights

else

print "no need to turn on lights yet"

if temperature greater than 30 then

turn on air conditioner

else

print "temperature is comfortable. Ac not needed"

close scanner

End

**Algorithm: steps**

1. Start the program.
2. Initialize the scanner to get user input.
3. Ask and read the current time (as an integer).
4. Ask and read the current temperature (as an integer).
5. If the time is exactly 19 (7 PM), turn on the lights.
6. Else, print a message that lights are not needed.
7. If the temperature is greater than 30°C, turn on the air conditioner.
8. Else, print that AC is not needed.
9. Close the scanner.
10. End the program.

**Code** :

import java.util.Scanner;

public class smarthome {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter current time: ");

int time = scanner.nextInt();

System.out.print("Enter current temperature in °C: ");

int temperature = scanner.nextInt();

if (time == 19) {

turnOnLights();

} else {

System.out.println("No need to turn on lights yet.");

}

if (temperature > 30) {

turnOnAC();

} else {

System.out.println("Temperature is comfortable. AC not needed.");

}

scanner.close();

}

public static void turnOnLights() {

System.out.println("Lights are ON.");

}

public static void turnOnAC() {

System.out.println("Air Conditioner is ON.");

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The Smart Home program uses user inputs for time and temperature to simulate basic automation.
* It turns on the lights if the time is exactly 19 (7 PM) and activates the air conditioner if the temperature exceeds 30°C. The logic is simple, clear, and demonstrates effective use of conditional statements to model real-world smart home behavior.

**SECTION 2 (The Tools for Comparison and Logic)**

**1.**  **Problem Statement** :Declare an int variable myAge and assign your age to it

Write expressions using comparison operators to check if

myAge is equal to 25

myAge is greater than 18.

myAge is less than or equal to 65.

myAge is not equal to 30.

Print the Boolean result of each expression using System.out.println()

**Pseudo code** **:**

Start

create scanner to read input

ask "enter your age"

read age as integer

check if age is equal to 25 print result

check if age is greater than 18 print result

check if age is less than or equal to 65 print result

check if age is not equal to 30 print result

close scanner

End

**Algorithm: steps**

1. Start the program.
2. Create a Scanner to take user input.
3. Prompt the user to enter their age.
4. Read the input and store it in myAge.
5. Print whether myAge == 25.
6. Print whether myAge > 18.
7. Print whether myAge <= 65.
8. Print whether myAge != 30.
9. Close the Scanner.
10. End the program.

**Code** :   
import java.util.Scanner;

public class agechecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your age: ");

int myAge = scanner.nextInt();

System.out.println("Is myAge equal to 25? " + (myAge == 25));

System.out.println("Is myAge greater than 18? " + (myAge > 18));

System.out.println("Is myAge less than or equal to 65? " + (myAge <= 65));

System.out.println("Is myAge not equal to 30? " + (myAge != 30));

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The program checks a user's age against multiple conditions using comparison operators.
* It clearly demonstrates how to evaluate and print Boolean results in Java, making it useful for learning basic decision-making and logical expressions.

**2.**  **Problem Statement** :Declare two String variables: username = "admin" and password = "password123".

Declare two more variables, enter username and enter Password, and assign some test values.

Write a logical expression that returns true only if both username and password match.

**Pseudo code** **:**

Start

set correct username = "admin"

set correct password = "password123"

ask user to enter username

read entered username

ask user to enter password

read entered password

if entered username equals correct username and entered password equals correct password then

print "login successful!"

else

print "invalid username or password"

close scanner

End

**Algorithm: steps**

1. Start the program.
2. Store the correct username and password.
3. Prompt the user to input username.
4. Read and store entered username.
5. Prompt the user to input a password.
6. Read and store the entered password.
7. Compare both entered values with the correct credentials.
8. If both match, print "Login successful!".
9. Else, print "Invalid username or password".
10. Close the Scanner.
11. End the program.

**Code** :

import java.util.Scanner;

public class LoginValidation {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String correctUsername = "admin";

String correctPassword = "password123";

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

String enteredUsername = scanner.nextLine();

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

String enteredPassword = scanner.nextLine();

if (correctUsername.equals(enteredUsername) && correctPassword.equals(enteredPassword)) {

System.out.println("Login successful!");

} else {

if (!correctUsername.equals(enteredUsername)) {

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

}

if (!correctPassword.equals(enteredPassword)) {

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

}

}

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The program checks login credentials using string comparison and logical operators.
* It effectively demonstrates how to validate user input and control access, making it a good example of basic authentication logic in Java.

**3.**  **Problem Statement** :Declare an int variable num and assign it a value.

Check whether num is:

Greater than 10 AND less than 20.

Less than 5 OR greater than 100.

Print the results

**Pseudo code** **:**

Start

create scanner to read user input

ask "enter a number"

read num

check if num > 10 and num < 20 → store in isbetween10and20

check if num < 5 or num > 100 → store in islessthan5orgreaterthan100

print result of first condition

print result of se

**Algorithm: steps**

1. Start the program.
2. Create a Scanner object.
3. Prompt the user to enter a number.
4. Read the input and store it in num.
5. Check if the number is greater than 10 **AND** less than 20.
6. Check if the number is less than 5 **OR** greater than 100.
7. Print both Boolean results.
8. Close the Scanner.
9. End the program.

**Code** :

import java.util.Scanner;

public class NumberRangeCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int num = scanner.nextInt();

boolean condition1 = (num > 10 && num < 20);

boolean condition2 = (num < 5 || num > 100);

if (condition1) {

System.out.println("The number satisfies: Greater than 10 AND less than 20.");

} else if (condition2) {

System.out.println("The number satisfies: Less than 5 OR greater than 100.");

} else {

System.out.println("The number does not satisfy any condition.");

}

scanner.close();

}

}

| Test cases | Input /Expected Output |
| --- | --- |
| TC1 |  |
| TC2 |  |
| TC3 |  |

**Observation**

* The program evaluates a number using logical operators (&&, ||) to check if it falls within or outside specific ranges.
* It clearly demonstrates how to use compound conditions and print Boolean results in Java.

**4. Activity** :Given the expression: 5+3\*2>10 &&!(7 ==7)

Break it down step-by-step.

Show the result after each stage of the operation and determine its

final Boolean value.

5+3\*2>10 &&!(7 ==7)

5+3\*2>10 &&!(TRUE)

5+3\*2>10 && FALSE

(8)\*2>10 && FALSE

(8\*2)>10 && FALSE

16 >10 && FALSE

(16 >10 )&& FALSE

TRUE && FALSE

FALSE

**SECTION 3 (Conditional Statements)**

**1.**  **Problem Statement** : Get an integer input from the user using Scanner

Write an if-else if-else structure that

o Prints "Positive" if the number is greater than 0 and the number is less than 0.

o Prints "Negative" if the o Prints "Zero' if the number is exactly 0.

**Pseudo code** **:**

Start

create scanner to read input

ask user to enter an integer

read number

if number > 0 then

print "positive"

else if number < 0 then

print "negative"

else

print "zero"

close scanner

End

**Algorithm: steps**

1. Start the program
2. Create a Scanner for input
3. Prompt the user to enter an integer
4. Read and store the number
5. If number > 0, print "Positive"
6. Else if number < 0, print "Negative"
7. Else, print "Zero"
8. Close the Scanner
9. End the program

**Code** :

import java.util.Scanner;

public class NumberCheck {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = scanner.nextInt();

if (number > 0) {

System.out.println("Positive");

} else if (number < 0) {

System.out.println("Negative");

} else {

System.out.println("Zero");

}

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The program takes an integer input and uses if-else statements to check whether the number is positive, negative, or zero.
* It clearly demonstrates basic number classification using conditional logic.

**2.**  **Problem Statement** : Ask the user to input their age.

Use an if-else structure to determine if they are eligible to drive

(age >= 18)

Problem 2.3: Simple Calculator

Get two double inputs and an operator (+-,\*,/) from the user

Use if-else if-else to perform the operation.

Handle division by zero using an if check.

**Pseudo code** **:**

Start

create scanner to read input

ask user to enter age

read age

if age >= 18 then

print "you are eligible to drive"

else

print "you are not eligible to drive"

close scanner

End

**Algorithm: steps**

1. Start the program
2. Create a Scanner for input
3. Prompt user to enter their age
4. Read and store age
5. If age ≥ 18, print "Eligible to drive"
6. Else, print "Not eligible to drive"
7. Close the Scanner
8. End the program

**Code** :

import java.util.Scanner;

public class DrivingEligibility {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your age: ");

int age = scanner.nextInt();

if (age >= 18) {

System.out.println("You are eligible to drive.");

} else {

System.out.println("You are not eligible to drive.");

}

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The program checks if a user's age is 18 or above to determine driving eligibility.
* It uses a simple if-else condition to demonstrate decision-making based on numeric input.

**3.**  **Problem Statement** :Get two double inputs and an operator (+-,\*,/) from the user

Use if-else if-else to perform the operation.

Handle division by zero using an if check.

**Pseudo code** **:**

Start

create scanner to read input

ask user for first number

read num1

ask user for second number

read num2

ask user for an operator (+, -, \*, /)

read operator

if operator is '+'

print num1 + num2

else if operator is '-'

print num1 - num2

else if operator is '\*'

print num1 \* num2

else if operator is '/'

if num2 ≠ 0

print num1 / num2

else

print "error: cannot divide by zero"

else

print "invalid operator"

close scanner

End

**Algorithm: steps**

1. Start the program
2. Create a Scanner for input
3. Ask and read first number (num1)
4. Ask and read second number (num2)
5. Ask and read operator (+, -, \*, /)
6. Use if-else to perform the corresponding operation
7. For division, check if num2 ≠ 0 to avoid divide-by-zero error
8. Print the result or error message
9. Close the Scanner
10. End the program

**Code** :

import java.util.Scanner;

public class SimpleCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the first number: ");

double num1 = scanner.nextDouble();

System.out.print("Enter the second number: ");

double num2 = scanner.nextDouble();

System.out.print("Enter an operator (+, -, \*, /): ");

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

if (operator == '+') {

System.out.println("Result: " + (num1 + num2));

} else if (operator == '-') {

System.out.println("Result: " + (num1 - num2));

} else if (operator == '\*') {

System.out.println("Result: " + (num1 \* num2));

} else if (operator == '/') {

if (num2 != 0) {

System.out.println("Result: " + (num1 / num2));

} else {

System.out.println("Error: Cannot divide by zero.");

}

} else {

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

}

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |
| TC4 |  |  |

**Observation**

* The program performs basic arithmetic operations based on user input.
* It uses if-else conditions to handle different operators and includes a check to prevent division by zero, demonstrating safe and interactive calculator logic.

**4.**  **Problem Statement** : Get user age (int) and student status (boolean).

Use nested if or logical operators to determine:

o If under 5 or over 65. $5

o If 5-18 and student: $8

o Otherwise: $12

Print the result.

**Pseudo code** **:**

Start

create scanner to read input

ask user for age

read age

ask if user is a student (true/false)

read isstudent

if age < 5 or age > 65 then

ticketprice = 5

else if age between 5 and 18 and isstudent is true then

ticketprice = 8

else

ticketprice = 12

print ticket price

close scanner

End

**Algorithm: steps**

1. Start the program
2. Create a Scanner for input
3. Ask and read the user's age
4. Ask and read whether the user is a student (boolean)
5. If age is less than 5 or greater than 65, set price to $5
6. Else if age is between 5 and 18 and student is true, set price to $8
7. Else, set price to $12
8. Print the ticket price
9. Close the Scanner
10. End the program

**Code** :

import java.util.Scanner;

public class MovieTicketPrice {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your age: ");

int age = scanner.nextInt();

System.out.print("Are you a student? (true/false): ");

boolean isStudent = scanner.nextBoolean();

int ticketPrice;

if (age < 5 || age > 65) {

ticketPrice = 5;

} else if (age >= 5 && age <= 18 && isStudent) {

ticketPrice = 8;

} else {

ticketPrice = 12;

}

System.out.println("Your ticket price is: $" + ticketPrice);

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The program calculates movie ticket prices based on age and student status using conditional logic.
* It demonstrates the use of logical operators and if-else statements to apply different pricing rules effectively.

**SECTION 4 (Switch Logic - handling multiple choice)**

**1.**  **Problem Statement** : Ask the user to input an integer from 1-7.

Use a switch statement to print the corresponding day.

Include a default case for invalid inputs.

**Pseudo code** **:**

Start

create scanner to read input

ask user to enter a number from 1 to 7

read day

switch (day)

case 1: print "sunday"

case 2: print "monday"

case 3: print "tuesday"

case 4: print "wednesday"

case 5: print "thursday"

case 6: print "friday"

case 7: print "saturday"

default: print "invalid input. Please enter a number from 1 to 7."

close scanner

End

**Algorithm: steps**

1. Start the program
2. Create a Scanner object
3. Ask the user to enter a number between 1 and 7
4. Read the number as day
5. Use a switch statement to check the value:

* If 1 → print "Sunday"
* If 2 → print "Monday" up to
* If 7 → print "Saturday"

1. Otherwise, print "Invalid input"
2. Close the Scanner
3. End the program

**Code** :

import java.util.Scanner;

public class DayOfWeek {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number (1 to 7) to get the day of the week: ");

int day = scanner.nextInt();

switch (day) {

case 1:

System.out.println("Sunday");

break;

case 2:

System.out.println("Monday");

break;

case 3:

System.out.println("Tuesday");

break;

case 4:

System.out.println("Wednesday");

break;

case 5:

System.out.println("Thursday");

break;

case 6:

System.out.println("Friday");

break;

case 7:

System.out.println("Saturday");

break;

default:

System.out.println("Invalid input. Please enter a number from 1 to 7.");

}

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The program maps numbers 1 to 7 to days of the week using a switch statement.
* It demonstrates clear control flow and handles invalid input with a default case, making it a good example of menu or option-based selection logic.

**2.**  **Problem Statement** : Simulate an ATM.

Get user input: 1 = Check Balance, 2 = Withdraw, 3 = Deposit, 4 =

Exit.

Use switch to print the action.

Handle invalid input with a default case.

**Pseudo code** **:**

Start

initialize balance to 1000

display menu options

ask user to choose an option (1-4)

read choice

switch (choice)

case 1: display current balance

case 2:

ask for withdrawal amount

if amount <= balance then

deduct amount from balance

display new balance

else

display "insufficient balance"

case 3:

ask for deposit amount

add amount to balance

display new balance

case 4:

display "thank you! Exiting"

default:

display "invalid option"

close scanner

End

**Algorithm: steps**

1. Start the program
2. Set initial balance to $1000
3. Show user a menu with 4 options
4. Take user input for selected option
5. Use switch to handle the choice:

* Case 1: Show balance
* Case 2: Ask withdrawal amount → check if sufficient → update balance
* Case 3: Ask deposit amount → update balance
* Case 4: Exit message
* Default: Invalid input message

1. Close the Scanner

End the program

**Code** :

import java.util.Scanner;

public class SimpleATM {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

double balance = 1000.00;

System.out.println("1. Check Balance");

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

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

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

System.out.print("Choose an option (1-4): ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.println("Your balance is: $" + balance);

break;

case 2:

System.out.print("Enter amount to withdraw: ");

double withdrawAmount = scanner.nextDouble();

if (withdrawAmount <= balance) {

balance -= withdrawAmount;

System.out.println("Withdrawal successful. New balance: $" + balance);

} else {

System.out.println("Insufficient Balance.");

}

break;

case 3:

System.out.print("Enter amount to deposit: ");

double depositAmount = scanner.nextDouble();

balance += depositAmount;

System.out.println("Deposit successful. New balance: $" + balance);

break;

case 4:

System.out.println("Thank you! Exiting");

break;

default:

System.out.println("Invalid option. Please choose between 1 and 4.");

}

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |

**Observation**

* The program simulates basic ATM operations using a switch statement.
* It allows checking balance, withdrawing, and depositing money, and includes input validation, demonstrating simple banking logic with user interaction.

**3.**  **Problem Statement** : Grade Remarks (Why switch is not ideal)\*\*is not ideal)

Input score (0-100).

Use if-else if-else to print:

ο 90-100: "Excellent"

80-89: "Very Good"

ο 70-79: "Good"

o 60-69: "Pass"

oBelow 60: "Fail"

Explain why switch would not be appropriate here

**Pseudo code** **:**

Start

create scanner to read input

ask user to enter score (0–100)

read score

if score between 90 and 100 then

print "excellent"

else if score between 80 and 89 then

print "very good"

else if score between 70 and 79 then

print "good"

else if score between 60 and 69 then

print "pass"

else if score between 0 and 59 then

print "fail"

else

print "invalid score"

close scanner

End

**Algorithm: steps**

1. Start the program
2. Create a Scanner for input
3. Ask the user to enter a score
4. Read and store the score
5. Use if-else if to check score ranges:

* 90–100 → "Excellent"
* 80–89 → "Very Good"
* 70–79 → "Good"
* 60–69 → "Pass"
* 0–59 → "Fail"
* Otherwise → "Invalid score"

1. Close the Scanner
2. End the program

**Code** :

import java.util.Scanner;

public class GradeRemarks {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your score (0-100): ");

int score = scanner.nextInt();

if (score >= 90 && score <= 100) {

System.out.println("Excellent");

} else if (score >= 80 && score <= 89) {

System.out.println("Very Good");

} else if (score >= 70 && score <= 79) {

System.out.println("Good");

} else if (score >= 60 && score <= 69) {

System.out.println("Pass");

} else if (score >= 0 && score < 60) {

System.out.println("Fail");

} else {

System.out.println("Invalid score. Please enter a number between 0 and 100.");

}

scanner.close();

}

}

| Test cases | Input | Expected Output |
| --- | --- | --- |
| TC1 |  |  |
| TC2 |  |  |
| TC3 |  |  |
| TC4 |  |  |

**Observation**

* The program evaluates a student's score and displays a remark based on defined grade ranges using if-else if statements.
* It effectively handles valid and invalid input, making it useful for understanding condition-based grading logic.