1. package day6\_assignment;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

// Enum for Suit

enum Suit {

CLUBS, DIAMONDS, HEARTS, SPADES

}

// Enum for Rank

enum Rank {

ACE, TWO, THREE, FOUR, FIVE, SIX, SEVEN,

EIGHT, NINE, TEN, JACK, QUEEN, KING

}

class Card {

private final Suit suit;

private final Rank rank;

public Card(Suit suit, Rank rank) {

this.suit = suit;

this.rank = rank;

}

public Suit getSuit() {

return suit;

}

public Rank getRank() {

return rank;

}

@Override

public String toString() {

return rank + " of " + suit;

}

}

// Deck class

class Deck {

private final List<Card> cards = new ArrayList<>();

public Deck() {

// Generate all 52 cards

for (Suit suit : Suit.values()) {

for (Rank rank : Rank.values()) {

cards.add(new Card(suit, rank));

}

}

}

public void shuffle() {

Collections.shuffle(cards);

}

public void printDeck() {

for (Card card : cards) {

System.out.println(card);

} }

}

public class Cards\_enum {

public static void main(String[] args) {

Deck deck = new Deck();

System.out.println("Original deck:");

deck.printDeck();

System.out.println("\nShuffled deck:");

deck.shuffle();

deck.printDeck();

}

}

2. package day6\_assignment;

class Animal {

void makeSound() {

System.***out***.println("Some generic animal sound");

}

}

class Dog extends Animal {

*@Override*

void makeSound() {

System.***out***.println("Woof!");

}

void fetch() {

System.***out***.println("Dog fetches the ball");

}

}

public class Castingdemo {

public static void main(String[] args) {

Dog d = new Dog();

Animal a = d;

a.makeSound(); // Prints: Woof!

// a.fetch(); // ❌ Compile-time error: Animal has no fetch() method

((Dog) a).fetch();

}

}

3. package day6\_assignment;

import java.util.Scanner;

enum *Direction* {

***NORTH***, ***SOUTH***, ***EAST***, ***WEST***

}

public class Compass\_directions\_enum{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter a direction (NORTH, SOUTH, EAST, WEST): ");

String input = scanner.nextLine().toUpperCase();

try {

*Direction* dir = *Direction*.*valueOf*(input);

// Use switch to print movement

switch (dir) {

case ***NORTH***:

System.***out***.println("Move north");

break;

case ***SOUTH***:

System.***out***.println("Move south");

break;

case ***EAST***:

System.***out***.println("Move east");

break;

case ***WEST***:

System.***out***.println("Move west");

break;

}

} catch (IllegalArgumentException e) {

System.***out***.println("Invalid direction entered. Please use NORTH, SOUTH, EAST, or WEST.");

}

scanner.close();

}

}

4. package day6\_assignment;

public class CompoundAssignmentDemo {

public static void main(String[] args) {

int x = 5;

// 1. Direct assignment with different types:

// x = x + 4.5; // ❌ Compile-time error!

// Reason: x + 4.5 promotes x to double (widening), result is double.

// Assigning double → int without explicit cast is illegal in Java

// because it’s a narrowing conversion that could lose data.

// 2. Compound assignment:

x += 4.5; // ✅ Compiles!

// Behind the scenes: x = (int)(x + 4.5);

// Compound assignment automatically casts back to int.

// The fractional part (.5) is truncated, so result is 9.

System.***out***.println(x); // Prints: 9

}

}

5. package day6\_assignment;

import java.util.Scanner;

enum *DaysOfWeek* {

***MONDAY***, ***TUESDAY***, ***WEDNESDAY***, ***THURSDAY***, ***FRIDAY***, ***SATURDAY***, ***SUNDAY***

}

public class Day\_of\_week\_enum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter a day of the week: ");

String input = scanner.nextLine().toUpperCase();

try {

*DaysOfWeek* day = *DaysOfWeek*.*valueOf*(input);

System.***out***.println("Position (ordinal) of " + day + " is: " + day.ordinal());

switch (day) {

case ***SATURDAY***:

case ***SUNDAY***:

System.***out***.println(day + " is a weekend!");

break;

default:

System.***out***.println(day + " is a weekday.");

}

} catch (IllegalArgumentException e) {

System.***out***.println("Invalid day entered.");

}

scanner.close();

}

}

6. package day6\_assignment;

enum *Difficulty* {

***EASY***, ***MEDIUM***, ***HARD***

}

class Game {

private int bullets;

private *Difficulty* difficulty;

public Game(*Difficulty* diff) {

this.difficulty = diff;

switch (diff) {

case ***EASY*** -> bullets = 3000;

case ***MEDIUM*** -> bullets = 2000;

case ***HARD*** -> bullets = 1000;

}

}

public void start() {

System.***out***.println("Game started with difficulty: " + difficulty);

System.***out***.println("You have " + bullets + " bullets!");

}

}

public class Difficult\_level\_game\_setup\_enum {

public static void main(String[] args) {

Game easyGame = new Game(*Difficulty*.***EASY***);

easyGame.start();

Game mediumGame = new Game(*Difficulty*.***MEDIUM***);

mediumGame.start();

Game hardGame = new Game(*Difficulty*.***HARD***);

hardGame.start();

}

}

7. package day6\_assignment;

public class Exceptiondemo {

public static void main(String[] args) {

// Division by zero demo

try {

int a = 10, b = 0;

int result = a / b;

System.***out***.println("Result: " + result);

} catch (ArithmeticException e) {

System.***out***.println("Division by zero is not allowed!");

} finally {

System.***out***.println("Operation completed.");

}

System.***out***.println(); // separator

// Array index out of bounds demo

try {

int[] numbers = {1, 2, 3};

System.***out***.println("Accessing 5th element: " + numbers[4]);

} catch (ArrayIndexOutOfBoundsException e) {

System.***out***.println("Array index out of bounds! Please use a valid index.");

} finally {

System.***out***.println("Operation completed.");

}

}

}

8. package day6\_assignment;

import java.util.Scanner;

public class Implicit\_widening {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Scanner inputScanner = new Scanner(System.***in***);

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

int originalInt = inputScanner.nextInt();

double widenedDouble = originalInt;

System.***out***.println("Original integer: " + originalInt);

System.***out***.println("Widened double: " + widenedDouble);

}

}

9. package day6\_assignment;

public class Int\_string\_conversion {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int originalInt = 12345;

System.***out***.println("Original int: " + originalInt);

String intAsString = String.*valueOf*(originalInt);

System.***out***.println("Converted to String: " + intAsString);

try {

int parsedInt = Integer.*parseInt*(intAsString);

System.***out***.println("Converted back to int: " + parsedInt);

String invalidString = "abc";

System.***out***.println("\nAttempting to parse invalid string: \"" + invalidString + "\"");

int invalidParsedInt = Integer.*parseInt*(invalidString);

System.***out***.println("This line will not be reached if exception occurs.");

} catch (NumberFormatException e) {

System.***err***.println("Error: NumberFormatException occurred. The string could not be parsed as an integer.");

System.***err***.println("Exception message: " + e.getMessage());

}

}

}

10. package day6\_assignment;

enum *PriorityLevel* {

***LOW***(1, "Low priority - minor issue"),

***MEDIUM***(2, "Medium priority - moderate impact"),

***HIGH***(3, "High priority - requires attention"),

***CRITICAL***(4, "Critical priority - immediate action needed");

private final int severity;

private final String description;

PriorityLevel(int severity, String description) {

this.severity = severity;

this.description = description;

}

public int getSeverity() {

return severity;

}

public String getDescription() {

return description;

}

public boolean isUrgent() {

return severity >= 3; // HIGH and CRITICAL are urgent

}

}

public class Prority\_level\_enum {

public static void main(String[] args) {

System.***out***.println("Priority Levels:");

System.***out***.println("----------------");

for (*PriorityLevel* level : *PriorityLevel*.*values*()) {

System.***out***.println(level +

" (Severity: " + level.getSeverity() +

") -> " + level.getDescription() +

" | Urgent? " + level.isUrgent());

}

}

}

11. package day6\_assignment;

import java.util.Scanner;

public class Read\_input\_user {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Scanner scanner = new Scanner(System.***in***);

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

int number = scanner.nextInt();

System.***out***.println("You entered: " + number);

}

}

12. package day6\_assignment;

import java.util.Scanner;

public class Temperature {

public static void main(String[] args) {

Scanner sc = new Scanner(System.***in***);

// Prompt the user

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

double celsius = sc.nextDouble();

double fahrenheit = celsius \* 9 / 5 + 32;

int fahrenheitTruncated = (int) fahrenheit;

System.***out***.println("Fahrenheit (precise): " + fahrenheit);

System.***out***.println("Fahrenheit (truncated to int): " + fahrenheitTruncated);

}}

13. package day6\_assignment;

// State interface

interface State {

State next();

}

// TrafficLight enum implementing State

enum *TrafficLight* implements State {

***RED*** {

*@Override*

public State next() {

return ***GREEN***;

}

},

***GREEN*** {

*@Override*

public State next() {

return ***YELLOW***;

}

},

***YELLOW*** {

*@Override*

public State next() {

return ***RED***;

}

};

}

public class Traffic\_light\_enum {

public static void main(String[] args) {

State light = *TrafficLight*.***RED***; // Start at RED

System.***out***.println("Traffic Light Simulation:");

for (int i = 0; i < 6; i++) {

System.***out***.println("Step " + (i + 1) + ": " + light);

light = light.next();

}

}

}

14. package day6\_assignment;

import java.util.Scanner;

public class TypeCasting\_explicit {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Scanner scanner = new Scanner(System.***in***);

System.***out***.print("Enter a double value: ");

double originalDouble = scanner.nextDouble();

System.***out***.println("Original double: " + originalDouble);

int castToInt = (int) originalDouble;

System.***out***.println("Casted to int: " + castToInt);

short castToShort = (short) castToInt;

System.***out***.println("Casted to short: " + castToShort);

}

}