Worksheet: Classes and Objects (Kotlin)

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_

Instructions: Complete the following exercises to practice your understanding of classes, objects, and related concepts in Kotlin. Write your code in the space provided or in your IDE.

## Exercise 1: Classes and Object Instances

Create a class called Car with properties for color, number of doors, and whether it is electric. Add a method to change the color. Instantiate two different cars and print their details.

Code/Answer:

## Exercise 2: Constructors

Write a class Book with a primary constructor that takes a title and an optional author (default: "Unknown"). Create several Book objects using different constructor arguments and print their properties.

Code/Answer:

## Exercise 3: Initializer Block

Create a class Rectangle with properties for width and height. Use an init block to print the area when a new rectangle is created.

Code/Answer:

## Exercise 4: Multiple Constructors

Implement a class Circle with a primary constructor for radius, and secondary constructors for diameter and for a default unit circle. Print the area in each case.

Code/Answer:

## Exercise 5: Properties, Getters, and Setters

Make a class Person with a mutable property name. Add a custom getter that always returns the name in uppercase, and a custom setter that trims whitespace. Test it in main().

Code/Answer:

## Exercise 6: Member Functions

Add a function to the Person class that returns a greeting message using the person's name. Call this function from main().

Code/Answer:

## Exercise 7: Inheritance

Create a base class Animal with a function makeSound(). Derive two classes, Dog and Cat, that override makeSound() with appropriate sounds. Demonstrate polymorphism in main().

Code/Answer:

## Exercise 8: Interfaces

Define an interface Shape with a function area(). Implement this interface in Circle and Rectangle classes. Calculate and print the area for each.

Code/Answer:

## Exercise 9: Extension Functions

Write an extension function isEven() for the Int class that returns true if the number is even. Test it with several numbers.

Code/Answer:

## Exercise 10: Data Classes

Create a data class Player with properties name and score. Instantiate a few players, print them, and demonstrate the copy() function.

Code/Answer:

## Exercise 11: Pair and Triple

Use Pair to store a book and its author, and Triple to store a book, author, and year. Print the values.

Code/Answer:

## Exercise 12: Enum Classes

Define an enum class Day for days of the week. Add a property to indicate if it’s a weekend. Write a function that prints whether a given day is a weekend.

Code/Answer:

## Exercise 13: Singleton Object

Create an object MathUtils with a function to calculate the factorial of a number. Call this function from main().

Code/Answer:

## Exercise 14: Companion Objects

Write a class BankAccount with a companion object that keeps track of the total number of accounts created. Print the count after creating several accounts.

Code/Answer:

## Exercise 15: Packages and Organization

Organize your Animal, Shape, and Player classes into separate packages. Demonstrate importing and using them in a main file.

Code/Answer:

## Exercise 16: Visibility Modifiers

Create a class with public, private, and protected properties/methods. Try accessing them from outside the class and from a subclass to see the effect.

Code/Answer: