Detailed Notes on the PDF Content: *Classes and Objects in Java*

Introduction

- Java is a fully object-oriented programming language.
- Programs in Java are designed using **classes** and **objects**.
- A Java program is made up of classes, and every program must encapsulate its logic inside a class.

Class in Java

- **Definition**: A class is a template or blueprint for creating objects. It defines the properties (variables) and actions (methods) of objects.
- **Attributes**:
  - Represented by variables.
  - Define the state of the class.
- **Methods**:
  - Represent actions or functionality.
  - Define the behavior of the class.
- **Declaration Syntax**:

  ```
  [modifier] class ClassName {
      // class body
  }
  ```

  Example:

  ```
  public class Employee {
      // class code
  }
  ```

- **Components**:
  1. Field declarations.
  2. Constructor declarations.
  3. Method declarations.
  4. Instance block declarations.
  5. Static block declarations.

Object in Java

- **Definition**: An object is a basic runtime entity in an object-oriented programming language. It represents a real-world entity with properties (state) and actions (behavior).
- **Characteristics**:
  - **State**: Represented by instance variables.
  - **Behavior**: Represented by methods.
  - **Identity**: Unique name differentiates one object from another.
- **Key Features**:
  - Objects are instances of a class.
  - They take up memory space independently.
  - Creation process: instantiating an object.
  - Example: A "Book" object has properties like title, author, and actions like "open" or "close".

Difference Between Classes and Objects

Aspect	Class	Object
Definition	User-defined data type.	Instance of a class.
Memory	Does not occupy memory.	Occupies memory.
Manipulation	Cannot be manipulated directly.	Can be manipulated.
Purpose	Generates objects.	Provides functionality to a class.

Points to Remember

1. Objects are the core runtime entities in object-oriented systems.
2. You can create multiple objects from a single class.
3. Each object holds independent memory space.
4. The process of object creation is called **instantiation**.

MCQ Question Bank on *Classes and Objects in Java*

Basic Concepts

1. **Which of the following is a true statement?**
    - (a) Objects define classes.
    - (b) Classes define objects.
    - (c) Classes and objects are unrelated.
    - (d) Java does not support classes or objects.
    - *Answer**: (b)
2. **What is the process of creating an object called?**
    - (a) Declaration
    - (b) Instantiation
    - (c) Compilation
    - (d) Initialization
    - *Answer**: (b)
3. **Which of these is NOT a characteristic of an object?**
    - (a) State
    - (b) Identity
    - (c) Method
    - (d) Behavior
    - *Answer**: (c)

Classes

4. **A class in Java serves as a:**
    - (a) Blueprint for creating objects.
    - (b) Runtime entity.
    - (c) Memory location for objects.
    - (d) Method to encapsulate main logic.

- *Answer*: (a)

5. **Which of the following is the correct way to declare a class?**
   - (a) `class ClassName { }`
   - (b) `new ClassName { }`
   - (c) `public ClassName { }`
   - (d) `declare class ClassName { }`
   - *Answer*: (a)

Objects

6. **An object’s state is represented by:**
   - (a) Methods
   - (b) Variables
   - (c) Constructors
   - (d) Static blocks
   - *Answer*: (b)

7. **Which keyword is used to create an object in Java?**
   - (a) `create`
   - (b) `class`
   - (c) `new`
   - (d) `object`
   - *Answer*: (c)

8. **What differentiates one object from another?**
   - (a) Behavior
   - (b) Method
   - (c) Identity
   - (d) State
   - *Answer*: (c)

Miscellaneous

9. **What is the relation between classes and objects?**
   - (a) Classes are instances of objects.
   - (b) Objects are templates for classes.
   - (c) Classes encapsulate objects.
   - (d) Objects are instances of classes.
   - *Answer*: (d)

10. **Which of the following is NOT a component of a class?**
    - (a) Methods
    - (b) Constructors
    - (c) Static blocks
    - (d) Main memory allocation
    - *Answer*: (d)

Let me know if you'd like more questions or revisions!