**Class 9 Chapter 1**

**Principles of object oriented programming**

**A. Tick (✓) the Correct Answer**

1. **Which of the following is a pillar of Object-Oriented Programming?**
   * **c. Polymorphism** (✓)
2. **\_\_\_\_\_ allows multiple objects of different subclasses to be treated as objects of a single superclass.**
   * **d. Super class** (✓)
3. **Abstraction and Data hiding maintain the \_\_\_\_\_ of data as only necessary data is provided.**
   * **c. Security** (✓)
4. **Which of the following language is known as a user-friendly computer language?**
   * **c. High-Level Language** (✓)
5. **The wrapping up of data members and member methods into a single unit is called Encapsulation.**
   * **d. together** (✓)

**B. Fill in the Blanks**

1. **Procedure-Oriented Programming (POP)** has global data sharing of functions.
2. **Compiler/interpreter** converts source code into object code.
3. **Object-Oriented Programming (OOP)** divides the whole problem into smaller programs known as functions or methods.
4. Methods and **data members/variable** are enclosed within a unit called class.
5. C++ is an example of **Object-Oriented Programming Language (OOPL).**

**C. Short Answer Type Questions**

1. **Write the difference between POP and OOP.**

| **Feature** | **POP (Procedure-Oriented Programming)** | **OOP (Object-Oriented Programming)** |
| --- | --- | --- |
| Approach | Focuses on procedures (functions) | Focuses on objects |
| Data Handling | Uses global data sharing | Uses encapsulation (data hiding) |
| Security | Less secure | More secure due to abstraction |
| Example | C | C++, Java, Python |

1. **Define Polymorphism with a real-life example.**
   * **Polymorphism** means "many forms." It allows the same function or operator to behave differently based on the context.
   * **Example:** A **person** can act as a **teacher** in school, a **parent** at home, and a **customer** in a shop.
2. **Write any 2 disadvantages of Object-Oriented Programming.**
   * **Higher memory usage** due to object creation.
   * **More complex** compared to procedural programming.
3. **Write the difference between Polymorphism and Encapsulation.**

| **Feature** | **Polymorphism** | **Encapsulation** |
| --- | --- | --- |
| Definition | Ability of a function or object to take multiple forms | Wrapping data and methods into a single unit |
| Purpose | Reduces code duplication | Provides data security |
| Example | Function overloading, method overriding | Private variables in a class |

1. **Give a real-life example of a situation where polymorphism is used.**
   * A **car** has a method called move(), but different types of vehicles (bike, truck, plane) override it with their own implementation.
2. **Write the difference between Assembly language and Machine level.**

| **Feature** | **Assembly Language** | **Machine Level Language** |
| --- | --- | --- |
| Language Type | Low-level | Binary code (0s and 1s) |
| Readability | Uses mnemonics (MOV, ADD) | Not human-readable |
| Speed | Slower than machine code | Fastest execution |
| Example | MOV A, B | 10101100 00001101 |

1. **Write down two advantages of Polymorphism.**
   * **Code reusability**: Reduces redundancy by allowing the same function to behave differently.
   * **Scalability**: Makes code more adaptable to future changes.
2. **Write down any two disadvantages of Machine Level Language.**
   * **Difficult to understand and write** since it consists of binary numbers (0s and 1s).
   * **Not portable**, as machine code is specific to a particular processor.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***