EXPERIMENT – 03

Objective:

To understand the Object-Oriented Programming (OOP) concepts and basics of Java programming.

Theory:

Object-Oriented Programming (OOP) is a programming paradigm centered around objects. Objects are instances of classes, which encapsulate data (attributes) and behavior (methods). Java is an object-oriented programming language that adheres to these principles.

Key OOP Concepts:

Concept	Description
Class	Blueprint for creating objects; defines attributes and methods.
Object	Instance of a class representing real-world entity.
Encapsulation	Bundling data and methods into a class, promoting abstraction.
Inheritance	Subclass derives from superclass, enabling code reuse.
Polymorphism	Method takes multiple forms via overloading/overriding.

Basics of Java Programming:

Java supports:
- Primitive data types: int, float, double, boolean, char
- Variables: store data values, must be declared
- Operators: arithmetic, relational, logical
- Control Flow: if-else, switch, loops
- Methods: perform specific tasks
- Classes and Objects: organize code
- Packages: organize classes, avoid conflicts

Code:

```
public class Person {

   String name;
   int age;
   void displayInfo() {
      System.out.println("Name: " + name);
```

```
        System.out.println("Age: " + age);
    }
}
 class Main {
    public static void main(String[] args) {
        Person person1 = new Person();
        person1.name = "Alice";
        person1.age = 30;
        person1.displayInfo();
    }
}
```

Output:

Name: Alice
Age: 30

Coding Questions:

(a) Java program to create 'Person' class using constructor:

```
public class Person {

    String name;
    int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public void printDetails() {
        System.out.println("Name: " + name + ", Age: " + age);
    }

    public static void main(String[] args) {
        Person person1 = new Person("Alice", 25);
        Person person2 = new Person("Bob", 30);
        person1.printDetails();
        person2.printDetails();
    }
}
```

Output:
Name: Alice, Age: 25
Name: Bob, Age: 30

(b) Java program to use private variables and getter/setter:

```
public class Person {

   private String name;
   private int age;
   private String country;

   public String getName() { return name; }
   public void setName(String name) { this.name = name; }

   public int getAge() { return age; }
   public void setAge(int age) { this.age = age; }

   public String getCountry() { return country; }
   public void setCountry(String country) { this.country = country; }

   public void printDetails() {
      System.out.println("Name: " + name + ", Age: " + age + ", Country: " + country);
   }

   public static void main(String[] args) {
      Person person = new Person();
      person.setName("Charlie");
      person.setAge(28);
      person.setCountry("USA");
      person.printDetails();
   }
}
```

Output:
Name: Charlie, Age: 28, Country: USA