Assignment – 4th (Fundamentals of Java)

1. What is Programming Language?

Ans: - A programming language is a set of instructions, symbols, and syntax that are used to write computer programs. Programming languages play a crucial role in the development of software and applications and have enabled the creation of complex and sophisticated systems. Programming languages can be divided into two main categories: low-level languages and high-level languages. Low-level languages, such as Assembly, are closer to machine code and provide more direct control over the computer's hardware. High-level languages, such as Python or Java, are more abstract and easier for humans to read and write.

2. Why do we need a programming language?

Ans: - Programming languages are needed because they provide a way for humans to communicate instructions to computers. They serve as a bridge between human-understandable concepts and the underlying machine-understandable code.

In short, programming languages are essential for software development because they make it possible for humans to write and execute code on computers. They provide a way for us to express our ideas, solve problems, and build complex systems and applications.

3. What are the features of Java?

Ans: - Java is a widely used, class-based, object-oriented programming language that was designed to be portable and secure. Java is a feature-rich programming language that is well-suited to the development of complex, scalable, and secure applications. Some of its key features include:

- **Object-Oriented: -** Java is an object-oriented programming language, meaning that it allows for the creation and manipulation of objects. This feature makes it possible to model real-world scenarios and implement complex systems and applications.
- **Platform Independent: -** Java is compiled to bytecode, which is then executed by the Java Virtual Machine (JVM). This means that Java code can be run on any platform that has a JVM installed, making it highly portable.
- **Dynamic: -** Java is designed to support dynamic class loading, which enables new classes to be loaded and instantiated at runtime. This feature makes it possible to create dynamic, flexible applications.
- **Secure:** - Java has built-in security features such as automatic memory management, exception handling, and type checking, making it a secure choice for developing applications that handle sensitive information.
- **Robust:** - Java has robust error handling and automatic memory management, making it less prone to bugs and memory leaks.

4. What is an Object?

Ans: - An Object is an instance of a class. A class is a template or blueprint from which objects are created. So, an object is the instance of a class. An object has two main components:

1. **State: -** The state of an object represents the data that it holds, such as its attributes, properties, or variables.
2. **Behaviour: -** The behaviour of an object represents the actions it can perform, such as the methods or functions it can execute.

In object-oriented programming, objects are used to model real-world entities, and their state and behaviour are used to represent the data and functionality of those entities.

5. What is a Class?

Ans: - A class is a template or blueprint from which objects are created. In object-oriented programming, a class is a basic building block. A class in Java is a logical entity. A class defines the properties (data members) and methods (functions) that the objects of that class will have.

In general, classes are used to encapsulate data and behaviour, and to create objects that can be used to model real-world entities. By defining classes, it becomes possible to create reusable, modular code that can be easily extended and maintained over time.

6. Explain about the main() method in Java?

Ans: - The main() method in Java is the entry point for a Java application. It is the method that is executed when the program starts running. The main() method has a specific signature in Java, which is defined as follows:

```
public static void main(String[] args) {}
```

The ***`public`*** keyword makes the method accessible from anywhere in the program. The ***`static`*** keyword means that the method can be called without creating an instance of the class that contains it. The ***`void`*** keyword indicates that the method does not return any value. The ***`main()`*** method takes an array of strings as its argument, which is typically used to pass command-line arguments to the application. The ***`String[] args`*** argument is an array of strings that represent the command-line arguments passed to the program. The main() method is the starting point of every Java application. For example –

```
public class FirstClass {
        public static void main(String[] args) {
                System.out.println("Hello, World!");
        }
}
```