**Q1. what is programming language?**

**Answer:** A programming language is a formal language that specifies a set of instructions that can be used to produce various kinds of output. These instructions, known as code, can be used to create software, mobile applications, websites, and other digital systems. Programming languages provide a way for developers to express algorithms, data structures, and other logical processes in a way that can be interpreted and executed by a computer.

There are many different programming languages, each with its own syntax, structure, and set of features. Some of the most widely used programming languages include Java, Python, C, C++, JavaScript, and Ruby. Different programming languages are suited to different tasks and environments, and a software developer typically chooses a language based on the requirements of the project they are working on.

**Q2. why do we need programming language**?

**Answer:** We need programming languages for the following reasons:

To communicate with computers: Computers can only understand binary code, which is difficult for humans to work with. Programming languages provide a higher-level, more abstract way of communicating with computers that is easier for humans to understand and work with.

To automate tasks: By writing code, we can automate tasks that would otherwise require manual labour, increasing efficiency and accuracy.

To build complex systems: Programming languages allow us to create complex systems, such as software applications, websites, and databases, that would be difficult or impossible to build using only binary code.

To solve problems: Programming languages can be used to solve a wide range of problems, from simple mathematical calculations to complex data analysis and machine learning.

To create new technologies: Programming languages are the foundation of the technology industry and play a key role in the development of new technologies, such as artificial intelligence, virtual reality, and the Internet of Things.

**Q3. what are features of java?**

**Answer:** Java is a general-purpose, object-oriented programming language that is widely used for developing a variety of applications, including desktop applications, mobile applications, web applications, and enterprise applications. Some of the key features of Java include:

Object-Oriented: Java is an object-oriented programming language, which means it is based on the concept of objects, classes, and inheritance. This makes it easy to model real-world objects and organize code into reusable components.

Platform-Independent: Java code can run on any device that has a Java Virtual Machine (JVM) installed, making it platform-independent. This means that a Java program can run on Windows, Mac, Linux, and other operating systems without modification.

Multi-Threaded: Java provides built-in support for multi-threading, allowing developers to write code that can run multiple tasks concurrently. This is useful for creating applications that need to perform multiple tasks simultaneously, such as downloading data in the background while the user interacts with the application.

Robust and Secure: Java is designed to be a robust and secure programming language, with features such as automatic memory management and type safety. This helps to prevent common programming errors and makes it harder for attackers to exploit vulnerabilities in Java applications.

Large Community and Libraries: Java has a large and active community of developers, and there are many libraries and tools available for Java development. This makes it easier for developers to find solutions to common problems, and to build applications faster.

Dynamic and Extensible: Java is a dynamic and extensible programming language, with features such as dynamic class loading, reflection, and dynamic proxies. This makes it easy for developers to write code that can be extended or modified at runtime.

**Q4. what is an object?**

**Answer:** An object in object-oriented programming is an instance of a class. It is a self-contained unit that combines data and behaviour.

A class defines a blueprint for an object, including the data (also known as attributes or properties) and the behaviour (also known as methods) that the object can exhibit. An object is an instance of a class, created at runtime, that has its own unique set of values for the attributes defined by the class.

**Q5. what is a class?**

**Answer:** A class in object-oriented programming is a blueprint for creating objects (instances of the class). It defines a set of attributes (data) and methods (behaviour) that are common to a particular type of object.

A class acts as a template for creating objects, and each object created from the class has its own set of values for the attributes defined by the class. Classes can also inherit attributes and methods from other classes, which allows for the creation of complex object hierarchies.

**Q6. explain about main () method in java?**

**Answer:** The main method in Java is the entry point of a Java application. It is a static method that is called when the Java program starts. The main method is defined within a Java class, and has the following signature:

public static void main (String [] args) {

// program logic goes here

}

The public keyword indicates that the main method can be called from anywhere. The static keyword indicates that the method can be called without creating an instance of the class. The void keyword indicates that the method does not return a value.

The main method takes an array of strings as an argument, which are passed to the program when it is executed. The args array is commonly used to provide command-line arguments to the program.

The body of the main method contains the logic of the program. This is where you write the code that the program will execute.

In Java, the main method is required for a program to be executable. When a program is executed, the Java Virtual Machine (JVM) looks for the main method in the class specified on the command line and calls it.