1) What is a Programming language?

Ans: A programming language is a set of instructions used to write software programs. It allows programmers to communicate with computers by creating code that the computer can understand and execute. Programming languages define the structure, rules, and syntax for writing code. Programmers write code in a programming language using an IDE or text editor, and it is then translated or compiled into machine-readable instructions. Programming languages are important for developing software and utilizing the computational capabilities of computers.

2) Why do we need programming language?

Ans: Programming languages are necessary to communicate with computers. Just as we need a specific language to communicate with people, we need a particular language to speak with a computer. These languages are called programming languages. Programming languages are used to create applications for performing specific tasks, automating things, etc. It can be used to create desktop applications, operating systems, websites, mobile applications, etc. There are many reasons why someone would want to learn programming. It can help you advance in your current job, increase your marketability for new jobs, improve your productivity as an employee or leader, and even increase your income.

3) What are the features of Java?

Ans: The main features of Java::

- Simple: easy to learn and understand syntax.
- Object-oriented: everything in Java is an object.
- Portable: can run on multiple platforms.
- Platform-independent: compiled Java code can run on all platforms that support Java without the need for recompilation.
- Secured: has features such as no explicit pointers, a classloader, and a bytecode verifier.
- Robust: designed to eliminate certain types of programming errors.
- Interpreted: bytecode is translated on the fly to machine instructions.
- High-performance: includes features such as just-in-time compilation to improve performance.
- Multithreaded: supports multiple threads of execution within a single program.

4) What is an object?

Ans: An object is an instance of a class. A class defines the characteristics and behaviours of the objects that are created from it. These characteristics are represented by attributes or properties, and behaviours are represented by methods.
An object is created by using the NEW keyword followed by the name of the class and any necessary parameters. Once an object is created, you can access its attributes and call its methods using dot notation.

5) What is a class?

Ans: A class is a blueprint for creating objects. It defines the characteristics and behaviours of the objects that are created from it.
A class is defined using the class keyword followed by the name of the class. The body of the class is enclosed in curly braces and contains the definitions of the attributes and methods of the class.

Attributes, also known as properties or fields, represent the characteristics or state of the objects created from the class. They are defined using variable declarations.
Methods represent the behaviours or actions that objects created from the class can perform. They are defined using function declarations.
Here's an example of a simple Car class that has two attributes (make and model) and one method (honk):

```
class Car {
        String make;
        String model;

        Car(String make, String model) {
         this.make = make;
         this.model = model;
        }

   void honk() {
        System.out.println("Beep beep!");
        }
}
```

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

Ans: The main() method in Java is the entry point of any Java application. It is the first method that gets called when a Java program is executed. The signature of the main() method is always the same and looks like this:

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

}
```

The meaning of each word is as follows:

- public: This means that the main() method is accessible from anywhere, including outside of the class where it is defined.
- static: This means that the main() method belongs to the class and not to any particular instance of the class. This is necessary because the main() method is called before any objects are created.
- void: This means that the main() method does not return any value.
- main: This is the name of the method.
- String[] args: This is an array of String values that represents the command-line arguments passed to the program when it is executed.