1. Explain the differences between primitive and reference data types.

   Primitive datatypes are basic and defined types of data. They are predefined by the programming language. Reference data types are complex data structures that store a reference (or address) to the memory location where the data is stored.

2. Define the scope of a variable (hint: local and global variable)

   The scope of a variable refers to the region of a program where the variable is defined and can be accessed. Local variables are located inside a function, a method or a block and it can only be accessed within that block. Global variables are declared outside a block usually at the top of the program file and can be accessed anywhere in the program.

3. Why is initialization of variables required.

   Initialization is required to allocate adequate memory space that will be used when the programming is executed.

4. Differentiate between static, instance and local variables.

   Static variables are declared using ‘static’ key word in a class but outside a method or constructor. Instance variables are variables declared within a class but outside methods and constructors and they lack the ‘static’ key word. Local variables are located inside a function, a method or a block and it can only be accessed within that block

5. Differentiate between widening and narrowing casting in java.

   Widening casting refers to the conversion of a smaller data type to a larger data type. Narrowing casting refers to the conversion of a larger data type to a smaller data type.

6. the following table shows data type, its size, default value and the range. Filling in the missing values.

TYPE	**SIZE (IN BYTES)**	**DEFAULT**	**RANGE**
boolean	1 bit	false	true, false
Char	2	\u0000	‘\0000’ to ‘\ffff’
Byte	1 byte	0	-128 to 127
Short	2 bytes	0	-2^{15} to $+2^{15}-1$
Int	4	0	-2,147,483,648 to 2,147,483,647
Long	8 bytes	0L	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Float	4	00.0f	±3.40282347E+38F
Double	8	0.0	-1.8E+308 to +1.8E+308

7. Define class as used in OOP

   A class is a blueprint of an object

8. Explain the importance of classes in Java programming.

   Classes facilitate encapsulation, which is the bundling of data (fields) and methods that operate on the data within a single unit (object). Encapsulation helps in hiding the internal state of an object from outside interference and misuse, thus promoting better security and maintainability of code.

   Classes support abstraction by providing a simplified view of the complex systems. Abstraction allows programmers to focus on the essential details of an object's behavior while hiding irrelevant details.

   Java classes support inheritance, where one class (subclass or derived class) can inherit attributes and behaviors from another class (superclass or base class).

   Polymorphism allows objects of different classes to be treated as objects of a common superclass.

   Classes facilitate code reusability through inheritance and composition. Inheritance allows subclasses to inherit and reuse behaviors and attributes from super-classes.