Q 1. what is garbage collection?

Garbage collection is a process in programming where the system automatically removes unused memory so your program can run efficiently.
Garbage collection removes objects that are no longer being used.

Java memory mainly uses:

- Heap → objects
- Stack → method calls, local variables

Garbage collection cleans heap memory**.**
Garbage collection handle by JVM.
Removes unreferenced objects.

Q.2: What are packages in Java?

A package in Java is a folder that stores related classes, interfaces together.
There are two types of packages:

1. Build in packages.
   Provided by Java.
   Examples: java.lang,
   java.util,java.util.

2. Userdefined packages.

   Created by developers.

Usually written in lowercase and reverse domain style.
Groups related classes.
Avoids naming conflicts.
Improves project structure.
Works like folders.
Q.3 What is the default package?

The default package is the package used when programmer don't write any package statement in a Java file.

So the class automatically belongs to the default package.

Q 4. Explain the use of import statements.

Import statement in Java is used **to** use classes from another package without writing their full package name every time.

Without import we must wrtite full class name.

```
	Eg. java.util.Scanner sc = new java.util.Scanner(System.in);
```

With import statement:

```
	import java.util.Scanner;

	Scanner sc = new Scanner(System.in);
```

Import improves readability

It does not increase memory usage

It avoids writing fully qualified names

Java.lang package is imported automatically

Q5. What are nested classes in Java?

A nested class in Java is a class defined inside another class.

It helps organize code when one class is only useful to another class.

There are two main categories.

1. Non-static nested class (Inner class)
An inner class needs an object of the outer class.

```
	class Outer {
   class Inner { void
      show() {
         System.out.println("Inner class");
      }
   }
}

public class Test {
   public static void main(String[] args) {
      Outer   o   =   new   Outer();
      Outer.Inner i = o.new Inner();
      i.show();
   }
}
```

2. Static nested class
Does not need an outer object.

```
	class Outer { static
   class Inner {
      void show() {
            System.out.println("Static nested class");
        }
    }
}

public class Test {
    public static void main(String[] args) {
        Outer.Inner i = new Outer.Inner();
        i.show();
    }
}
```

1. Write a program to sort characters in a String alphabetically.

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

                    String str = "nitin";
                    char[] ch = str.toCharArray();

 for (int i = 0; i < ch.length - 1; i++) {
for (int j = i + 1; j < ch.length; j++) {
                        if (ch[i] > ch[j]) {
                            char temp = ch[i];
                            ch[i] = ch[j];
                            ch[j] = temp;
                        }
                    }
                }

                    System.out.println("Sorted String: " + new String(ch)); }
            }
```

2.Write a program to convert String to char array.

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

                                str = "Hello";

char[] arr = str.toCharArray();
 for (char c : arr) {
System.out.println(c);
                                }
                            }
                        }
```

3.Write a program to find the length of a String without using length().

```
            public class StringLength {
```

```
    public static void main(String[] args) {
        String str = "Hello"; char[] ch
        = str.toCharArray(); int count
        = 0;
        for (char c : ch) {
            count++;
        }

        System.out.println("Length of string: " + count); }
}
```

4. Write a program to replace a character in a String.

```
public class ReplaceChar { public static
    void main(String[] args) {
        String str = "hello";
        String result = str.replace('l', 'x');
        System.out.println("Original: " + str);
        System.out.println("Updated: " + result);
    }
}
```

5. Write a program to compare two Strings without using

   equals().

```
          public class CompareStrings {
    public static void main(String[] args) {
        String str1 = "hello";
        String str2 = "hello";

        boolean isEqual = true;

        if (str1.length() != str2.length()) {
            isEqual = false;
        } else { for (int i = 0; i < str1.length();
            i++) {
if (str1.charAt(i) != str2.charAt(i)) {
                isEqual = false; break;
            }
            }
        }

        if (isEqual) {
            System.out.println("Strings are equal");
        } else {
System.out.println("Strings are not equal");
        }
    }
}
```