1. What are keywords in Java? List 10 commonly used keywords.

1)super

2)this

3)public

4)for

5)extends

6)implements

7)return

8)void

9)int

10)while

2. What is operator precedence? How does it affect the outcome of expressions?

Operator Precedence:

it determines the order in which operators are evaluated in an expression.

Operators with higher precedence are evaluated before lower ones.

Operator Precedence Affects Expressions in Java:

Operator precedence affects how an expression is evaluated, especially

when multiple operators are involved without parentheses.

int result = 10 + 5 \* 2;

System.out.println(result);

it prints 20 instead of 30 because \* has high precedence than + .

Order of operator precedence:

() – Parentheses

++, -- – Unary

\*, /, %

+, -

3. What are the types of operators in Java?

Type of operators

Arithmetic operators: +, -, \*, /, %

Relational operators: ==, !=, <, >, <=, >=

Logical operators: &&, ||

Assignment operators: =, +=, -=, /=, \*=

Unary operators: +, -, ++, --, !

Bitwise operator: <<, >>

Ternary operators: ? :

4.What are the different primitive data types available in Java?

In java,there are 8 data types. They are:

byte – A small 8-bit signed integer with a range from -128 to 127.

short – A 16-bit signed integer used for saving memory in large arrays.

int – A 32-bit signed integer and the most commonly used whole number type.

long – A 64-bit signed integer used when int is not large enough

float – A 32-bit decimal number with less precision, used for saving memory.

double – A 64-bit decimal number, offering more precision than float.

char – A 16-bit data type used to store a single unicode character.

boolean – A data type that holds only two values: true or false.

5.What is the default value of each primitive data type in Java?

data\_type default value

byte 0

short 0

int 0

long 0

float 0.0f

double 0.0d

character '\u000'

boolean false

6.Explain the difference between primitive and non-primitive data types in Java.

(a) Primitive Data Types:

Definition: These are the basic, built-in data types provided by Java.

Types: byte, short, int, long, float, double, char, and boolean.

Size and Structure: Fixed size, memory-efficient, and store simple values.

Operations: Perform operations directly using operators (+, -, etc.).

Default Values: Have default values (e.g., 0 for int, false for boolean).

Not Objects: They are not objects and don’t have methods.

Stored in: Stored in the stack memory.

(b) Non-Primitive Data Types:

Definition: These are data types created using classes and interfaces.

Examples: String, Arrays, Classes, Interfaces, Wrapper classes, etc.

Size and Structure: Can be large, complex, and dynamic in size.

Operations: Require methods to operate (.length(), .equals(), etc.).

Default Values: Default value is null.

Are Objects: They are objects and have methods and properties.

Stored in: Stored in the heap memory (reference in stack).

7. What is the size and range of each primitive data type in Java?

Data Type Size Range

byte 1 byte -128 to 127

short 2 bytes -32,768 to 32,767

int 4 bytes -2,147,483,648 to 2,147,483,647

long 8 bytes -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

float 4 bytes \~±3.4e−38 to ±3.4e+38 (approximate)

double 8 bytes \~±1.7e−308 to ±1.7e+308 (approximate)

char 2 bytes 0 to 65,535 (Unicode characters)

boolean 1 bit true or false

2. Explain the purpose of the following keywords: static, final, this, super.

static:

we can access any static field by using class name

Belongs to the class rather than any object.

Can be used with variables, methods, blocks, or nested classes.

Shared by all instances.

final:

Used to declare constants, prevent method overriding or inheritance.

Cannot be changed once assigned.

this:

Refers to the current object of the class.

Used to access instance variables and methods.

super:

Refers to the parent class (superclass) of the current object.

Used to call superclass methods or constructors.

8. What is the use of this and super in method overriding?

this keyword:

Refers to the current class instance.

Used to access current class's variables or methods.

super keyword:

Refers to the parent class.

Used to call the parent class method or constructor of the superclass.

9. Explain wrapper classes and their use in Java.

Wrapper classes are object representations of primitive data types.

Primitive Wrapper Class

byte -> Byte

short -> Short

int -> Integer

long -> Long

float -> Float

double -> Double

char -> Character

boolean -> Boolean

10.What are control statements in Java? List the types with examples.

Control Statements:

Control statements in Java are used to control the flow of execution in

a program based on conditions or loops.

There are 3 types of control statements.

(1)Decision-Making Statements

a)if statement

int age=20;

if(age>18){

System.out.println("Adult");

}

b)if else statement

int num = 5;

if (num % 2 == 0) {

System.out.println("Even");

} else {

System.out.println("Odd");

}

c)ladder if else

int marks = 85;

if (marks >= 90) {

System.out.println("Grade A");

} else if (marks >= 75) {

System.out.println("Grade B");

} else {

System.out.println("Grade C");

}

d)switch statement

int day = 2;

switch (day) {

case 1: System.out.println("Sunday"); break;

case 2: System.out.println("Monday"); break;

default: System.out.println("Invalid day");

}

2)Looping statements

a)for loop

for(int i=1;i<4;i++){

System.out.println(i);

}

b)while loop

int i=1;

while(i<4){

System.out.println(i);

i++;

}

c)do-while loop

int i=1;

do{

System.out.println(i);

i++;

}while(i<4);

3)Jumping Statements

a)break

for(int i=1;i<4;i++){

if(i==2)break;

System.out.println(i);

}

b)continue

for(int i=1;i<4;i++){

if(i==2)continue;

System.out.println(i);

}

c)return

public static int square(int x) {

return x \* x;

}

11. What is the difference between break and continue statements?

Both break and continue are jump statements used inside loops or switch statements to

alter the normal flow of control, but they behave very differently.

Break statement:

break is used to exit the loop or switch block immediately.

Ex:

for (int i = 1; i <= 5; i++) {

if (i == 3) {

break;

}

System.out.print(i);

}

output: 1 2

Continue statement:

continue is used to skips the current iteration of the loop and moves to the next iteration.

Ex:

for (int i = 1; i <= 5; i++) {

if (i == 3) {

continue;

}

System.out.print(i);

}

output: 1 2 4 5

12. Explain the differences between while and do-while loops with examples.

Both while and do-while loops are used to execute a block of code repeatedly,

but they differ in when the condition is checked.

While loop:

The condition is checked first, before executing the loop body.

Loop may not run even once if the condition is false.

Ex:

int i = 1;

while (i < 5) {

System.out.print(i);

i++;

}

output: 1 2 3 4

Do while loop:

The loop body is executed first, then the condition is checked.

The loop executes at least once, even if the condition is false.

Ex:

int i = 1;

do {

System.out.print(i);

i++;

} while (i < 5);

output: 1 2 3 4 5

13. What is the difference between char and String in Java?

char :

char is a primitive datatype

it stores single character

char has 2 bytes of memory

syntax : char a='H';

String:

String is a wrapper class and non primitive datatype.

it can stores sequence of characters, object

object stored in heap

it is immutable

syntax: String s="Hello";

14. Compare and contrast for, while, and do-while loops.

for loop:

for is used at the beginning for condition check.

it is best for when number of iterations is known.

for can execute 0 times if condition fails.

syntax: for(initialization,condition,increment)

while loop:

while is used at the beginning for condition check.

it is best for when number of iterations depend on condition.

while can execute 0 times if condition fails.

syntax: while(condition)

do-while loop:

while is used at the beginning for condition check.

it is best for When at least one execution is required.

do-while can execute 1 time even if condition fails.

syntax: do{}while(condition)