Q. 1. Explain the concept of data abstraction and inheritance.

- Java is a high-level class-based, Object-Oriented Programming language that is

designed to have as few Implementation dependencies as possible.

- Data abstraction is the reduction of a particular body of data to a simplified representation of the whole.
- Abstraction, in general, is the process of removing characteristics from something to reduce it to a set of essential elements.
- To this end, data abstraction creates a simplified representation of the underlying data, while hiding its complexities and associated operations.
- In computing, data abstraction is commonly used in object-oriented programming (OOP) and when working with a database management system (DBMS).
- Inheritance is the passing on of genetic traits from parents to their offspring, and these offspring get all the genetic information from their parents.

- Single Inheritance

When a Derived Class to inherit properties and behavior from a single Base Class , it is called as single inheritance.

- Multi Level Inheritance

A derived class is created from another derived class is called Multi Level Inheritance

- Hierarchical Inheritance

More than one derived classes are created from a single base class, is called Hierarchical Inheritance .

- Hybrid Inheritance

Any combination of above three inheritance (single, hierarchical and multi level) is called as hybrid inheritance.

- Multipath inheritance

Multiple inheritance is a method of inheritance in which one derived class can inherit properties of base class in different paths. This inheritance is not supported in.

- Multiple Inheritance

Multiple inheritances allows programmers to create classes that combine aspects of multiple classes and their corresponding hierarchies.

Q. 2. What are keywords? Explain.

Java Reserved Keywords

- Java reserved keywords are predefined words, which are reserved for any

functionality or meaning.

- We can not use these keywords as our identifier names, such as class name

or method name. These keywords are used by the syntax of Java for some

functionality.

- In the Java programming Language , a keyword is any one of 67 reserved

words that have a predefined meaning in the language.

- If we use a reserved word as our variable name, it will throw an error.

In Java, every reserved word has a unique meaning and functionality.

Consider the below syntax:

1. double marks;

in the above statement, double is a reserved word while marks is a valid identifier.

Below is the list of reserved keywords in Java:

abstract continue for protected transient

assert default goto public try

boolean do if static throws

break double implements strictfp package

byte else import super private

case enum interface short switch

catch extends instanceof return void

char final int synchronized volatile

class finally long throw date

const float native this while

Q. 3. Discuss the History of Java in details.

♦ Java is a high-level class-based, Object-Oriented Programming language that is

designed to have as few Implementation dependencies as possible.

♦ Java is like C++ Programming. C++ is an Immediate descendent of C language.

Every innovation in language design was driven by the need to take care of crucial

issues that the first language couldn t̰ tackle.

♦ Java was originally developed by James Gosling at Sun Microsystems. It was

released in May 1995 as a core component of Sun Microsystems' Java platform.

♦ James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language

project in June 1991. The small team of sun engineers called Green Team.

♦ The history of Java starts with the Green Team. Java team members

(also known as

Green Team), initiated this project to develop a language for digital devices such as

set-top boxes, televisions, etc.

♦ The language was initially called ‗Oak ‗after an oak tree that stood outside Gosling's

office. Later the project went by the name Green and was finally renamed Java, from

Java coffee, a type of coffee from Indonesia

♦ Java Derives a lot of its character from C and C++.The Java Creators realized that

utilizing the well-known grammar of C and reverberating the Object Oriented

Features of C++ would make their Language appealing to C/C++ developers. The

difference between the way Java and other programming languages worked was

revolutionary.

♦ By the late 1990s Java had brought multimedia to the Internet and started to grow

beyond the Web, powering consumer devices (such as cellular telephones), retail and

financial computers, and even the onboard computer of NASA Ş Mars exploration

rovers.

♦ Sun Microsystems released the first public implementation as Java 1.0 in 1996

♦ The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly

with the Java 1.0 language specification.

♦ Because of this popularity, Sun created different varieties of Java for different

purposes, including Java SE for home computers, Java ME for embedded devices,

and Java EE for Internet servers and supercomputers.

♦ In 2010 the Oracle Corporation took over the management of Java when it acquired

Sun Microsystems.

Q. 4. Explain Java tokens in short.

1. Often you might have seen large Java applications with thousands of lines of codes, but have you ever wondered what lies at its core? Well, these are tokens, the smallest individual elements, also known as the building blocks of a Java program.

2. A java Program is made up of Classes and Methods and in the Methods are the Container of the various Statements And a Statement is made up of Variables, Constants, operators etc .

3. Tokens are the various Java program elements which are identified by the compiler and separated by delimiters. The delimiters are not part of

the tokens. A token is the smallest element of a program that is meaningful to the compiler. The compiler breaks lines into chunks of text called tokens. Tokens supported in Java include keywords, variables, constants, special characters, operations etc.

4. In Java, a program is a collection of classes and methods, while methods are a collection of various expressions and statements. Tokens in Java are the small units of code which a Java compiler uses for constructing those statements and expressions. Java supports 5 types of tokens which are:

Keywords

Identifiers

Literals

Operators

Special Symbols

5. Keywords

Keywords in Java are predefined or reserved words that have special

meaning to the Java compiler. Each keyword is assigned a special task or function and cannot be changed by the user. You cannot use keywords as variables or identifiers as they are a part of Java syntax itself. A keyword should always be written in lowercase as Java is a case sensitive language.

6. Identifier

Java Identifiers are the user-defined names of variables, methods, classes, arrays, packages, and interfaces. Once you assign an identifier in the Java program, you can use it to refer the value associated with that identifier in later statements.

7. Literals

Literals in Java are similar to normal variables but their values cannot be changed once assigned. In other words, literals are constant variables with fixed values. These are defined by users and can belong to any data type. Java supports five types of literals which are as follows:

Integer

Floating Point

Character

String

Boolean

8. Operators

An operator in Java is a special symbol that signifies the compiler to perform some specific mathematical or non-mathematical operations on one or more operands. Java supports 8 types of operators.

Arithmetic + , –, / , * , %

Unary ++ , —, !

Assignment = , += , -= , *= , /= , %= , ^=

Relational ==, != , < , >, <= , >=

Logical && , ||

Ternary (Condition) ? (Statement1) : (Statement2);

Bitwise & , | , ^ , ~

Shift << , >> , >>>

9. Special Symbols

Special symbols in Java are a few characters which have special meaning known to Java compiler and cannot be used for any other purpose.

Brackets [] These are used as an array element reference and also indicates single and multidimensional subscripts

Parentheses() These indicate a function call along with function parameters

Braces{} The opening and ending curly braces indicate the beginning and end of a block of code having more than one statement

Comma (,) This helps in separating more than one statement in an expression

Semi-Colon (;) This is used to invoke an initialization list

Asterisk (*) This is used to create a pointer variable in Java

10. Separators: Separators are the lines that are used to virtual group related items together.

Q. 5. What are literals? Explain briefly.

1. What Are Literals in Java?

Variables are critical in programming because that s how you store data in a particular memory location. For example, a Java program, while running, stores values in containers known as vàriables, which are defined as a basic storage unit. To enhance the program s readability, the programmer must follow particular conventions while naming these variables and assigning values to them. For example, a source code representing a fixed value is "literal".

2.Literals in Java are a synthetic representation of boolean, character, numeric, or string data. They are a means of expressing particular values within a program. They are constant values that directly appear in a program and can be assigned now to a variable. For example, here is an integer variable named ƭcount assigned as an integer value in this statement:

```
int count = 0;
```

3. Types of Literals in Java

Literals in Java are typically classified into six types and then into various sub-types. The primary literal types are:

a. Integral Literals

b. Floating-Point Literals

c. Char Literals

d. String Literals

e. Boolean Literals

f. Null Literals

4. Integral Literals

Integral literals consist of digit sequences and are broken down into these sub-types:

Decimal Integer: Decimal integers use a base ten and digits ranging from 0 to 9. They can have a negative (-) or a positive (+), but non-digit characters or commas aren t allowed between characters. Example: 2022, +42, -68.

Octal Integer: Octal integers use a base eight and digits ranging from 0 to 7. Octal integers always begin with a 0. Example: 007, 0295.

Hexa-Decimal: Hexa-decimal integers work with a base 16 and use digits from 0 to 9 and the characters of A through F. The characters are case-sensitive and represent a 10 to 15 numerical range. Example: 0xf, 0xe.

Binary Integer: Binary integers uses a base two, consisting of the digits 0 and 1. The prefix 0b represents the Binary system. Example: 0b11011.

5. Floating-Point Literals

Floating-point literals are expressed as exponential notations or as decimal fractions. They can represent either a positive or negative value, but if it s not specified, the value defaults to positive. Floating-point literals come in these formats:

Floating: Floating format single precision (4 bytes) end with an f or F. Example: 4f. Floating format double precision (8 bytes) end with a d or D.

Example: 3.14d.

Decimal: This format uses 0 through 9 and can have either a suffix or an exponent. Example: 99638.440.

Decimal in Exponent form: The exponent form may use an optional sign, such as a "-," and an exponent indicator, such as "e" or "E." Example: 456.5f.

6. Char Literals

Character (Char) literals are expressed as an escape sequence or a character, enclosed in single quote marks, and always a type of character in Java. Char literals are sixteen-bit Unicode characters ranging from 0 to 65535. Example: char ch = 077.

7. String Literals

String literals are sequences of characters enclosed between double quote ("") marks. These characters can be alphanumeric, special characters, blank spaces, etc.

Examples: "John", "2468", "₩n", etc.

8. Boolean Literals

Boolean literals have only two values and so are divided into two literals:

True represents a real boolean value

False represents a false boolean value

So, Boolean literals represent the logical value of either true or false. These values aren't case-sensitive and are equally valid if rendered in uppercase or lowercase mode. Boolean literals can also use the values of 0̲ and 1̲.

9. Null Literals

Null literals represent a null value and refer to no object. Nulls are typically used as a marker to indicate that a reference type object isn t̲ available. They often describe an uninitialized state in the program. It is a mistake to try to dereference a null value. Example: Patient age = NULL;

10. How to Use Literals

Programmers who want to incorporate literals in their program begin with the prefix =, followed by the specific value.

Q. 6. Write a program to demonstrate the concept of data hiding and encapsulation.

???

Q. 7. Discuss any 3 loop holes in Java array creation.

1. An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

2. ooping is a feature that facilitates the execution of a set of instructions repeatedly untill a certain condition holds false. Java provides three types of loops namely the for loop, the while loop, and

the do-while loop. Loops are also known as Iterating statements or Looping constructs in Java.

3. Looping Constructs in Java are statements that allow a set of instructions to be performed repeatedly as long as a specified condition remains true.

A very obvious example of loops can be daily routine of programmers i.e. Eat -> Sleep -> Code -> Repeat

4. Need for Looping Constructs in Java

A computer is most suitable for performing repetitive tasks and can tirelessly do tasks tens of thousands of times. We need Loops because

Loops facilitates 'Write less, Do more' - We don't have to write the same code again and again.

They reduce the size of the Code.

Loops make an easy flow of the control.

They also reduce the Time Complexity and Space Complexity - Loops

are faster and more feasible as compared to Recursion.

5. Elements of the Loops in Java

Initialization Expression(s): Initialization is carried out only once before entering the loop. Here, we either declare and initialize a control variable(s) or only initialize the variable(s) we are going to use in looping. We can initialize multiple variables as well.

6. Test Expression (Condition): It is a boolean expression. Its value decides whether the loop will be executed or terminated. If the condition is satisfied, the control goes inside the loop, otherwise, it is terminated. In an exit-controlled loop, the test expression is evaluated before exiting from the loop whereas in an entry-controlled loop the test expression is evaluated before entering the loop.

7. Body of the Loop: The statements that are to be executed repeatedly are written in the body of the Loop. They are executed until the condition of the loop is true.

8. Update Expression(s): Here, we update the value of the loop

variable(s). It is used so that after some point of time the loop terminates. It is executed at the end of the loop when the loop-body has been executed and the next iteration is to start. It is also called Increment/Decrement expression since in most of the cases value of loop variables is either incremented or decremented.

9. Java has three types of loops i.e. the for loop, the while loop, and the do-while loop. for and while loops are entry-controlled loops whereas do-while loop is an exit-controlled loop.

10. The for Loop

When we know the exact number of times the loop is going to run, we use for loop. It provides a concise way of writing initialization, test condition, and increment/decrement statements in one line. Thus, it is easy to debug and also has no risk of forgetting any part of the loop, since the condition is checked before.

For loop is an entry-controlled loop as we check the condition first and then evaluate the body of the loop.

11. The while Loop

The while loop is used when the number of iterations is not known but the terminating condition is known. Loop is executed until the given condition evaluates to false. While loop is also an entry-controlled loop as the condition is checked before entering the loop. The test condition is checked first and then the control goes inside the loop.

Although for loop is easy to use and implement, there may be situations where the programmer is unaware of the number of iterations, it may depend on the user or the system. Thus, when the only iterating and/or terminating condition is known, while loop is to be used.

12. The do-while Loop

The do-while loop is like the while loop except that the condition is checked after evaluation of the body of the loop. Thus, the do-while loop is an example of an exit-controlled loop.

The loop is executed once, and then the condition is checked for further iterations. This loop runs at least once irrespective of the test condition,

and at most as many times the test condition evaluates to true. It is the only loop that has a semicolon(;).

Q. 8. List all the operators in Java and explain any 3 of them in detail.

1. Operators constitute the basic building block to any programming language, they are classified based on the functionality they provide.

2. Java too provides many types of operators which can be used according to the need to perform various calculations and functions.

3. Java supports the following types of operators:

Arithmetic Operators.

Assignment Operators.

Logical Operators.

Relational Operators.

Unary Operators.

Bitwise Operators.

Ternary Operators.

Shift Operators.

4. One of them is Arithmatic Operator,these operators involve the mathematical operators that can be used to perform various simple or advanced arithmetic operations on the primitive data types.

5. The Java programming language supports various arithmetic operators for all floating-point and integer numbers. These operators are + (addition), - (subtraction), * (multiplication), / (division), and % (modulo).

6. Logical operators are used to performing logical AND, OR and NOT operations, i.e. the function similar to AND gate and OR gate in digital electronics. They are used to combine two or more conditions/constraints or to complement the evaluation of the original condition under particular consideration. One thing to keep in mind is, while using AND operator, the second condition is not evaluated if the

first one is false. Whereas while using OR operator, the second condition is not evaluated if the first one is true, i.e. the AND and OR operators have a short-circuiting effect. Used extensively to test for several conditions for making a decision.

AND Operator (&&) –if(a && b) [if true execute else don t]

OR Operator (||) –if(a || b) [if one of them is true execute else don t]

NOT Operator (!) –!(a<b) [returns false if a is smaller than b]

7. Assigning a value to a variable seems straightforward enough; you simply assign the stuff on the right side of the '= 'to the variable on the left. Below statement 1 assigning value 10 to variable x and statement 2 is creating String object called name and assigning value "Amit" to it.

8. Primitive Assignment:

The equal (=) sign is used for assigning a value to a variable. We can assign a primitive variable using a literal or the result of an expression.

Q. 9. Write short note on Arrays with a suitable example.

1. An array is a variable that can store multiple values. For example, if you want to store 100 integers, you can create an array for it.

2. To create an array variable in C, a programmer specifies the type of the elements and the number of elements to be stored in that array.

3. This is called a single-dimensional array. The arraySize must be an integer constant greater than zero and type can be any valid data type.

4. An array is a data structure that contains a group of elements. Typically these elements are all of the same data type, such as an integer or string.

5.Arrays are commonly used in computer programs to organize data so that a related set of values can be easily sorted or searched.

6. For example, a search engine may use an array to store Web pages found in a search performed by the user. When displaying the results, the program will output one element of the array at a time. This may be done for a specified number of values or until all the values stored in the array have been output. While the program could create a new variable for each result found, storing the results in an array is much more efficient way to manage memory.

7. How to declare an array?

dataType arrayName[arraySize];

For example,

float mark[5];

Here, we declared an array, mark, of floating-point type. And its size is 5. Meaning, it can hold 5 floating-point values.

It's important to note that the size and type of an array cannot be changed once it is declared.

8. Access Array Elements

You can access elements of an array by indices.

Suppose you declared an array mark as above. The first element is mark[0], the second element is mark[1] and so on.

9. Few keynotes:

Arrays have 0 as the first index, not 1. In this example, mark[0] is the first element.

If the size of an array is n, to access the last element, the n-1 index is used. In this example, mark[4]

Suppose the starting address of mark[0] is 2120d. Then, the address of the mark[1] will be 2124d. Similarly, the address of mark[2] will be 2128d and so on.

This is because the size of a float is 4 bytes.

10. How to initialize an array?

It is possible to initialize an array during declaration. For example,

```
int mark[5] = {19, 10, 8, 17, 9};
```

You can also initialize an array like this.

```
int mark[] = {19, 10, 8, 17, 9};
```

Here, we haven't specified the size. However, the compiler knows its size is 5 as we are initializing it with 5 elements.

Q10.Explain the features of Java.

1. Inspired by C and C++

Java is inspired by C and C++. The syntax of Java is similar to these languages but the languages are quite different. Java inherits many features from C and C++. Compared to C++, Java code runs a bit slower but it is more portable and offers better security features.

2. Simple and Familiar

Java programming language is simple to learn, understand, read, and

write. Java programs are easy to create and implement compared to other programming languages such as C and C++. If you are familiar with the basic principles of programming or the concept of OOP (object-oriented programming), it would be easy to master Java.

3. Object-Oriented

Java is a fully object-oriented language, unlike C++ which is semi object-oriented. It supports every OOP concept such as Abstraction, Encapsulation, Inheritance, Polymorphism. Java programs are developed using classes and objects. Another notable feature is that in Java the main() function is defined under a class.

4. Platform Independent

Java ş platform independence means that Java programs compiled on one machine or operating system can be executed on any other machine or operating system without modifications. It is also called an Architecture Neutral Language.

Java supports WORA (Write Once, Run Anywhere), which means that programmers can develop applications in one operating system and run

on any other without any modification.

5. Dynamic

Java is more dynamic compared to C and C++. It can adapt to its evolving environment. It allows programmers to dynamically link new class libraries, objects, and methods. Java programs can have a large amount of run-time information that can be used to resolve accesses to objects.

6. Robust

Java is a robust language that can handle run-time errors as it checks the code during the compile and runtime. If any runtime error is identified by the JVM, it will not be passed directly to the underlying system. Instead, it will immediately terminate the program and stop it from causing any harm to the system. Java has a strong memory management system. It also supports the concepts of garbage collection and exception handling.

7. Secure

Java is a secure language that ensures that programs cannot gain access

to memory locations without authorization. It has access modifiers to check memory access. Java also ensures that no viruses enter an applet. Java ş bytecode verifier checks the code blocks for any illegal code that violates the access right. It also does not allow programmers to explicitly create pointers.

8. High Performance

Java offers high performance as it used the JIT (Just In Time) compiler. The compiler only compiles that method which is being called. The JIT enhances the performance of interpreting byte code by caching interpretations.

9. Portable

Due to the concept of Write Once Run Anywhere (WORA) and platform independence, Java is a portable language. By writing once through Java, developers can get the same result on every machine. It is also very portable to various operating systems and architectures. Java run-time system is written in the ANSI C with clean portability boundary which is POSIX-compliant.

10. Automatic Garbage Collection

The most common and critical problems in C/C++ were memory leaks. FYI, memory leaks are a problem wherein a piece of memory that is no longer in use, fails to be freed. This is because garbage collection has to be done manually. However, Java supports automatic Garbage collection. It keeps checking over such unused memory spaces and frees them automatically.