class pen { } ← class for pens

Pen p = new pen(),

↳ object/instance of the pen class

OOP

1. What is OOPS?

Answer:

OOPs stand for Object Oriented Programming System. It is a technique of programming that focuses on the object. Object means a real-world entity such as a pen, keyboard, chair, table, computer, Book etc. Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects. An Object can be defined as an instance of a class.

2. What are the main principles of OOPS?

Answer:

It's main principles include -

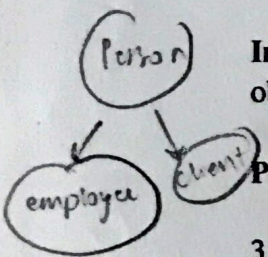
eg. A person class can have many properties / attributes But all these properties in the class should not be included in the main method - new Person p = Person(12)

↳ This is abstraction

Abstraction is hiding the complexity of the program that simplifies the program.

Encapsulation is binding the code and data and protecting it from being tampered.

↳ when we are using the Obj p we are not messing with the code inside



Inheritance refers reusability of the code where new object can inherit the property of parent object.

↳ lets say Person class { canWalk(); canTalk(); } class emp extends Person { canWork(), } has access to parent attributes / actions too

Person class

Polymorphism means having many forms.

Further it can have different forms too

3. What is a class?

↳ like a cookie cutter

Answer:

A class is a template from which objects are created. A class is a blueprint, or prototype which defines and describes the member attributes and member functions.

4. What is an object?

↳ cookie

Answer:

An object is an instance of a class. It has its own state, behavior, and identity.

Software objects are modeled after real-world objects such as a pen, keyboard, chair, table, computer, Book etc.

Just as Real-world objects have its state and behavior; Software objects too have state and behavior.

Better days are coming
keep believing!

A software object maintains its state in variables and implements its behavior with methods.

5. What is Encapsulation? Implementation

Answer:

Encapsulation is a principle of wrapping data (variables) and code together as a single unit. Encapsulation is like your bag in which you can keep your pen, book etc. Encapsulation means hiding the internal details of an object. Encapsulation is a technique used to protect the information in an object from the other object

6. What is Polymorphism? Implementation

Answer:

The word polymorphism means having many forms. It is the ability of a message to be displayed in more than one form. Like a man at a same time is a father, a husband, a employee. So, a same person posses different behavior in different situations. This is called polymorphism.

7. What is Inheritance?

class emp extends Person ? ?

Answer:

Inheritance facilitates reusability and is an important concept of OOPs. Just like a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class.

8. Define a constructor. → initialize state

name: " "
email: " "

Answer:

A constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation.

A constructor is an instance method that usually has the same name as the class, and can be used to set the values of the members of an object

9. Define Destructor.

Answer:

A destructor is a method which is automatically called when the lifetime of an object ends.

The purpose of the destructor is to free the resources that the object may have acquired during its lifetime

11. What is a virtual function? → used to override a behavior/method of derived class at runtime

Answer:

A virtual function is a member function which is declared within base class and is re-defined (Overriden) by derived class.

You use virtual functions when you want to override a certain behavior for your derived class at run-time.

12. What is a friend function? → A fun^n that can access private/protected data of a class

Answer:

If a function is defined as a friend function then the protected and private data of a class can be accessed using the function.

13. What is function overloading?

int add (int x, int y) { int s = x+y; return s; } (methods)

int add (int x, y, z) { int s = x+y+z; return s; }

Fun^n overloading

Answer:

Function overloading is a feature where two or more functions can have the same name but different parameters.

The function can perform different operations and hence eliminates the use of different function names for the same kind of operations.

Example
void Show(int& a, int& b);
void Show(double& a, double& b);
void Show(struct bob& a, struct bob& b);

14. What is operator overloading?

Answer:

It is a type of polymorphism in which an operator is overloaded to give user defined meaning to it. Overloaded operator is used to perform operation on user-defined data type.

15. What is an abstract class? → example

Answer:

Abstraction is a process of hiding the implementation details and showing only functionality to the user. Abstraction lets you focus on what the object does instead of how it does.

A class which is declared as abstract is known as an abstract class. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

16. What is an interface? → example

Answer:

Interfaces specify what a class must do and not how. It is the blueprint of the class.

An interface can have methods and variables, but the methods declared in interface are by default abstract

It is used to achieve total abstraction.

17. Difference between overloading and overriding.

Answer:

Method overloading is used to increase the readability of the program. Method overriding is used to provide the specific implementation of the method that is already provided by its super class. parent class walk(), and employee class walk().

Method overloading is the example of compile time polymorphism. Method overriding is the example of run time polymorphism.

The idea behind method overriding is to change the implementation of given method in a subclass. In other words you "override" the implementation of the parent's class method using the same signature of the method (name, return types, parameters), but implement different functionality inside the overridden method.

18. Difference between a class and an object.

Answer:

An instance of a class is known as an object. A template or blueprint with which objects are created is known as a Class.

Object allocates memory when it is created. Class doesn't allocate memory when it is created.

Object is a physical entity. Class is a logical entity.

19. What is an abstraction?

Abstraction is an important feature of OOPS, and it shows only the necessary details to the client of an object.

Abstraction is a process of hiding the implementation details from the user, only the functionality will be provided to the user. In other words, the user will have the information on what the object does instead of how it does.

20. What is early and late binding? → implementation

The compiler performs a process called binding when an object is assigned to an object variable. The early binding (static binding) refers to compile time binding and late binding (dynamic binding) refers to runtime binding.

While Early Binding, methods, functions and properties which are detected and checked during compile time and perform other optimizations before an application executes. The biggest advantage of using early binding is for performance and ease of development.

In Late binding functions, methods, variables and properties are detected and checked only at the run-time. It implies that the compiler does not know what kind of object or actual type of an object or which methods or properties an object contains until run time.

21. What is a copy constructor?

This is a special constructor for creating a new object as a copy of an existing object. There will always be only one copy constructor that can be either defined by the user or the system.

22. What is static and dynamic binding? Late binding

↑ early binding

Connecting a method call to the method body is known as binding.

When compiler acknowledges all the information required to call a function or all the values of the variables during compile time, it is called "static binding".

Calling a function or assigning a value to a variable, at run-time is called "Dynamic Binding"

Java Basics

1. How Java enabled High Performance?

Java uses Just-In-Time compiler to enable high performance.

Java source code is first compiled into byte code. These byte codes are then executed by JVM. But execution of byte code is slower than execution of machine language code, because JVM first needs to translate byte code into machine language code. Now here come Java JIT compiler that turns Java bytecode into instructions, compiling it "just in time" to run that can be sent directly to the processor.

2. Why Java is considered dynamic? Yes

Dynamic refer as the things which are executed when required rather than in advance. Java is considered as dynamic because of Bytecode. A source code written in one platform, the same code can be executed in any platform. Java is designed to adapt to an evolving environment. Java also loads the class files at runtime.

3. What is JIT compiler? JIT → code → Processor instructions. Hence, faster!

In Java, a just-in-time (JIT) compiler is a program that turns Java bytecode into instructions that can be sent directly to the processor. Just-In-Time(JIT) compiler is used to improve the performance.

4. Define class. → Blueprint of an object

A class is a template that is used to create objects. A class contains fields and methods to describe the behavior of an object.

5. What is Singleton class?

Singleton class controls object creation. It allows only one instance of itself to be created. This class is helpful when a user wants to restrict instantiation of a class to only one object. For example, if you want only one connection to a database due to licensing issues, you can use Singleton class in such scenario.

6. What is a static variable?

static variable is a class level variable and common to all the instances of the class. Only a single copy of static variable is created and shared among all the instances of the class.

It is declared with the static keyword in a class, but outside a method, constructor or a block.

7. What is protected access modifier?

Variables, methods and constructors which are declared protected in a superclass are accessible within package and outside the package but through inheritance only.

8. Why is String called immutable? → because created object cannot be changed.

The String class is considered as immutable because once it is created a String object cannot be changed. If we require to make a lot of modifications to Strings of characters then StringBuffer should be used.

9. Explain the importance of finalize() method.

- Finalize() method is fired when an object is just about to be reclaimed by the garbage collector. When an object needs to perform some action before the object is destroyed, Finalize() method is handy.

- Sometimes an object is holding some non-java resources that should be released before it is destroyed, finalization process of used.

10. Can we force the garbage collection to run?

- Yes, we can force but it is not certain that garbage collector will act upon immediately.
- We can use, System.gc() or runtime.gc()

11. What is the static method?

A static method belongs to the class rather than the object.

There is no need to create the object to call the static methods.

A static method can access and change the value of the static variable.

12. What is the purpose of 'this' keyword?

- 'this' keyword is used to refer current instance of an object.

- It is a non-static method and cannot be used inside main method in Java.

- It invokes current class constructor.

- It can be passed as an argument in the method call.

13. Significance of Super keyword.

- It is used to access the methods of a parent class.

- It is a non-static method and cannot be used inside main method in Java.

- It invokes the constructor of the parent class.

- The Outer.super can be used to get current instance of an outer class and its parent in Java.

14. How objects are stored in Java?

In java, each object gets a memory space from a heap. When an object is destroyed by a garbage collector, the space allocated to it from the heap is re-allocated to the heap and becomes available for any new objects.

15. What is Polymorphism in Java? What are the kinds of Polymorphism?

Polymorphism means the ability to take more than one form. It is the capability of an action or method to do different things based on the object acting upon.

Two types of Polymorphism:

1. Method Polymorphism through overloading

2. Object Polymorphism by inheritance and interface

16. What is method overloading?

Method overloading allows us to create multiple methods with the same name but different signature.

We used Method overloading when a couple of methods are needed with conceptually similar functionality with different parameters.

We can achieve method overloading in two ways:

By changing the number of arguments

By changing the return type

17. Can main() method in Java return any data?

In java, main() method is always declared with a void return type, so it can't return any data

18. Is it poosible to pass argument to a function by reference instead of pass by value in Java?

No, we can pass argument to a function only by value and not by reference in Java

19. Can a class have multiple constructors?

Yes, a class can have multiple constructors with different parameters. You can define as many constructors as you need. When a Java class contains multiple constructors, we say that the constructor is overloaded.

20. Is String a data type in java?

String is not a primitive data type in java. It's a class, a reference type. It is an object of Java.Lang.String class.

21. What is multi-threading?

Multi threading is running multiple tasks in a concurrent manner within a single program. Multi-threading is a special form of multitasking. Multi-threading allows you to write very efficient program that makes maximum use of CPU because idle time is kept to minimum.

22. Can we use Pointers in Java class?

We can't use concept of pointers in Java. But reference internally uses a pointer and does not allow manipulation to this

23. Can a class in Java be inherited from more than one class?

In Java, a class can be derived from only one class and not from multiple classes. Multiple inheritance is not supported by Java.

24. What is Garbage collector?

Garbage collector is the process of automatically freeing objects that are no longer referenced by the program. It reduces a programmer's work from having to track when to free allocated memory. Java provides memory management through garbage collector.

25. How are destructors defined in Java?

In Java, there are no destructors defined in the class as there is no need to do so. Java has its own garbage collection mechanism which does the job automatically by destroying the objects when no longer referenced.

26. Can we have static methods in an Interface?

Static methods can't be overridden in any class while any methods in an interface are by default abstract and are supposed to be implemented in the classes being implementing the interface. So it makes no sense to have static methods in an interface in Java.

Interface Methods are required to be implemented in the classes that implement the interface. But Static methods can't be overridden in any class. So, we can't have static methods in an Interface.

27. How does Java reduce the chances of a program going out of memory?

Java provides memory management through automatic garbage collection mechanism. This helps in reducing the chances of a program going out of memory. But it doesn't ensure that a Java program will not go out of memory. When Java objects are created at a faster pace compared to garbage collection, a program can go out of memory.

28. Can we increase the size of an array after its declaration?

No, Arrays are static, we can't change the size after its declaration. In this case, we should use vector over array.

29. What is the base class of all exception classes?

In Java, Java.lang.Throwable is the super class of all exception classes and all exception classes are derived from this base class.

30. What is the order of call of constructors in inheritance?

When we create a new object of a derived class, the constructor of the super class is invoked first and then the constructor of the derived class is invoked.

Java

1. What are the various access specifiers for Java classes?

The access modifiers in java specify accessibility (scope) of a data member, method, constructor or class. There are 4 types of java access modifiers:

Public: Class, Method, Field are accessible from anywhere.

Protected: The protected access modifier is accessible within package and outside the package but through inheritance only.

Default: If you don't use any modifier, it is treated as default by default. The default modifier

is accessible only within package.

Private: Methods and Fields are accessible only within class

2. What is the purpose of static methods and static variables?

We use static keyword to make a method or variable shared for all objects. When we want to have a single copy of a variable or method to be shared among all the instances of the class, we create static methods and static variables.

3. What do you mean by Checked Exceptions?

A checked exception is a subclass of Exception. A Compiler checks to see if these exceptions have been properly caught or not else the code doesn't compile.

Thus, a programmer is forced to deal with the situations where an exception can be thrown. Checked exceptions must be either declared or caught at compile time.

4. Explain the difference between an Inner Class and a Sub-Class.

An Inner class is a class which is nested within another class. An Inner class has access rights for the class which is nesting it and it can access all variables and methods defined in the outer class.

A sub-class is a class which inherits from another class called super class. Sub-class can access all public and protected methods and fields of its super class.

5. What is a singleton class?

A singleton class in java can have only one instance and hence all its methods and variables belong to just one instance. The purpose of singleton is to control the object creation, limiting the number of the objects to one only.

6. What is Final Keyword in Java?

In java, Final Keyword is used to define an entity that can only be assigned once.

Following is what happens when it is applied on classes, variables or methods:

final classes - A final class cannot have subclasses.

final variables - Once initialized, a final variable cannot be changed.

final Methods - A final method cannot be overridden by subclasses.

7. What are Java Packages?

eg: create a pkg person with methods like walk(), swim(); and variables and classes
and then import into a project that is organizing 'people in the world'

A package is a collection of classes and interfaces which are kept together as they are related to each other.

It helps modularizing the code and group the code for proper re-use.

Packages can be imported in other classes.

8. Explain the difference between an Abstract Class and Interface in Java.

cannot have method implementations

Abstract classes can have methods with implementation whereas an interface provides absolute abstraction and can't have any method implementations.

(A) abstract class:
→ can declare fields that are static, final
→ methods: public, protected and private can be defined

A class which implements an interface must implement all the methods of the interface while a class which inherits from an abstract class doesn't require implementation of all the methods of its super class.

Subclasses use extends keyword to extend an abstract class whereas subclasses use implements keyword to implement interfaces

(B) Interfaces:
→ all fields are auto public static final
→ methods: public

A class can implement multiple interfaces but it can extend only one abstract class.

9. Can we declare a class as Abstract without having any abstract method?

Yes, we can have an abstract class without abstract methods as both are independent concepts. Declaring a class abstract means it can't be instantiated on its own and can only be sub classed. Declaring a method abstract means method will be defined in the subclass.

However, if a class has even one abstract method, it must be declared as abstract otherwise it will give an error.

10. What is an immutable class?

Immutable class is a class which once created; its contents can't be changed.

Immutable objects are the objects whose state can't be changed once constructed.

Since the state of the immutable objects can't be changed once they are created, they are automatically synchronized/thread-safe.

abstract method:

```
abstract void moveTo (int a, int b);
```

abstract class:

```
public abstract class graphicObj {
    void move();
    abstract void draw();
}
```

→ Abstract class → class declared as abstract (can / ~~can~~ no abstract method included)
- cannot be instantiated (cannot create objects) but can be sub classed.

→ Abstract method → declared without an implementation
- methods that are abstract are in a class, it has to be declared as an abstract class

Where to use:

Abstract classes

→ Share code among several closely related classes

→ When you want children classes of the abstract classes to have different access modifiers.

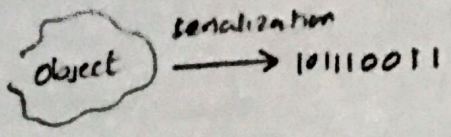
→ static / final non fields.

Interfaces

→ Implement unrelated classes.

→ specify behaviour

11. How an object is serialized in java?



Serialization is a mechanism of converting an object into a byte stream so that the object can be easily saved to persistent storage or streamed across a communication link. An object of a class is serialized by implementing serializable interface.

12. When we should use serialization?

Serialization is used when data needs to be transmitted over the network.

You can also save object's state and convert into byte stream using Serialization

13. When the constructor of a class is invoked?

The constructor of a class is invoked when an instance of the object is created and memory is allocated for the object.

For example, in the following class, two objects are created using new keyword and hence, constructor is invoked twice.

Example:

```
public class const_new {
 const_new() {
  system.out.println("constructor");
 }
 public static void main(String args[]) {
   const_new c1 = new const_new();
   const_new c2 = new const_new();
 }
}
```

14. Why Strings in Java are called as Immutable?

The String class is considered as immutable because once it is created, a String object cannot be changed. If we require to make a lot of modifications to Strings of characters then StringBuffer should be used.

15. Why Runnable Interface is used in Java?

Runnable interface is used for implementing multi threaded applications.

If you want to execute a code in separate thread, you need to implement Runnable interface in a class whose instance you want to run in separate thread instead of Main thread.

multi-threaded app

16. What are the two ways of implementing multi-threading in Java?

Java offers implementation of multi-threading by two ways:

By extending Thread class and overriding its run() method.

By Implementing Runnable interface

extend Thread class & overriding run()

implementing Runnable interface.

17. When a lot of changes are required in data, which one should be preferred - String or StringBuffer?

Since StringBuffers are dynamic in nature and we can change the values of StringBuffer objects unlike String which is immutable, it's always a good choice to use StringBuffer when data is being changed too many times. If we use String in such a case, for every data change a new String object will be created which will be an extra overhead.

18. How garbage collection is done in Java?

In java, when an object is not referenced any more, garbage collection takes place and the object is destroyed automatically. Java calls either System.gc() method or Runtime.gc() method for automatic garbage collection

19. How objects of a class are created if no constructor is defined in the class?

When no explicit constructor is defined in a java class, objects can still get created successfully. In this case, Java use default constructor implicitly for object creation. This constructor has no parameters.

20. How can we ensure that a resource isn't used by multiple threads simultaneously?

Using synchronized keyword, we can ensure that only one thread can use shared resource at a time and others can get control of the resource only once it is free from the other one using it.

21. How can we make copy of a java object?

Object cloning refers to creation of exact copy of an object. It creates a new instance of the class of current object and initializes all its fields with exactly the contents of the corresponding fields of this object.

Clone() is a method of Cloneable interface and hence, Cloneable interface needs to be implemented for making object copies.

22. What is the benefit of using inheritance?

Reusability - Inheritance offers facility to use public methods of base class without rewriting the same.

Extensibility - It extends the base class logic as per business logic of the derived class.

Data hiding - In inheritance, base class can decide to keep some data private so that it cannot be altered by the derived class

Overriding - With inheritance, we will be able to override the methods of the base class so that meaningful implementation of the base class method can be designed in the derived class.

23. What are two methods to resist a class from inherited in java?

1. By declaring the class as 'final'

2. By declaring all member's of the class private.

24. What is the difference between Stack and Queue?

A stack is based on Last in First out (LIFO) principle while a queue is based on FIFO (First In First Out) principle.

25. How can we restrict certain variables of a class from getting serialized?

restricting variables being serialized.

We can do so by using the keyword transient while declaring variables of the class.

For example, the variable trans_count below is a transient variable and can't be serialized:

Example

implements Serializable

```
public class transientSample {
   private transient trans_count;
}
```

26. Is it possible to use a default constructor even if an explicit constructor is defined?

When an explicit constructor is defined, default constructor can't be invoked. Java invokes default constructor when no explicit constructor is defined

27. Can we override a method by using same method name and arguments but different return types?

For a method to be overriden method name, arguments and return type must be exactly same. so, using a different return type doesn't override a method.

Data structures

1. What is data structure?

Answer:

A data structure refers to the format for organizing and storing data. It makes data access more efficient. Data structure types include the array, the file, the record, the table, the tree, and so on.

2. When is a binary search best applied?

Answer:

Binary search is a type of algorithm. It is best applied to search in a list in which the elements are already in order or sorted.

The binary search algorithm starts searching the list from the middle. If the middle value is not the correct one, then it will go on to search the top or the bottom half in a similar manner. The idea behind binary search itself is to cut down your search space by half after each comparison. And this is only possible if the dataset is sorted.

3. What is a linked list?

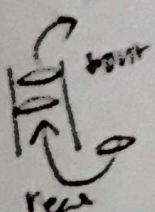
Answer:

A linked list is a sequence of nodes in which each node is connected to the node following it. A linked list consists of nodes where each node contains a data field and a reference to the next node in the list. The elements in a linked list are linked using pointers.

which are acting like pointers

4. What is a queue?

Answer:

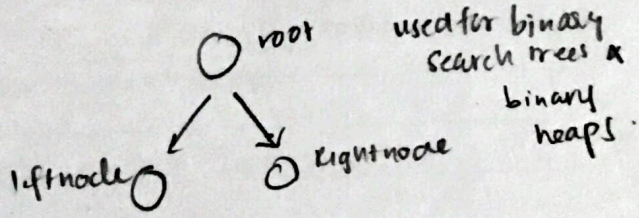


A queue is an ordered collection of items. In a queue, the addition of new items happens at one end, called the "rear" and the removal of existing items occurs at the other end,

commonly called the "front". As an element enters the queue, it starts at the rear and makes its way toward the front.

This makes queue as FIFO(First in First Out) data structure, which means that element inserted first will be removed first.

5. What are binary trees?

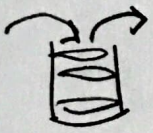


Answer:

A binary tree is one type of data structure that has two nodes, a left node, and a right node. It is made of nodes, where each node contains a "left" reference, a "right" reference, and a data element. The topmost node in the tree is called the root.

Binary trees are used to implement binary search trees and binary heaps

6. What is a stack?



Answer:

A stack is a data structure in which data is inserted and removed according to the last-in first-out (LIFO) principle, leaving the most recently added data on top. You can think of a stack of books; you can remove only the top book, also you can add a new book on the top.

7. Which data structures are applied when dealing with a recursive function?

What is recursion? a function that calls itself based on a terminating condition

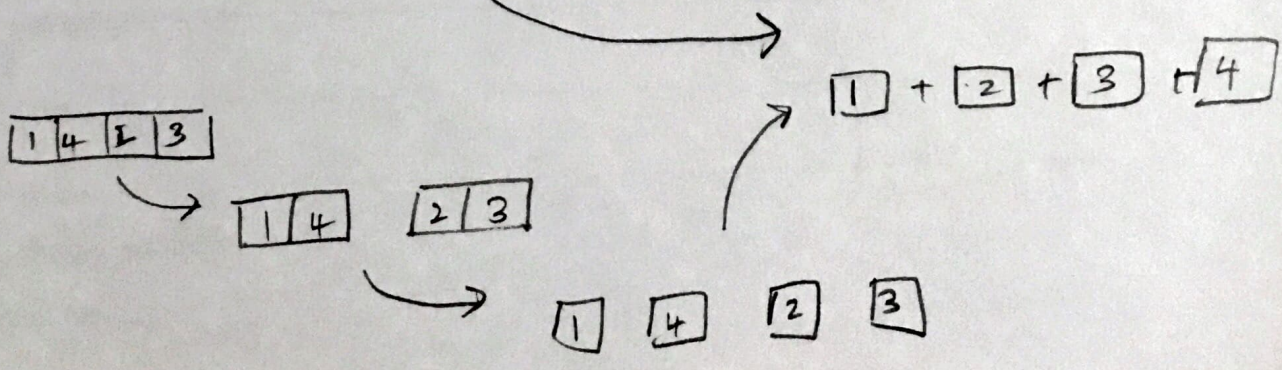
Answer:

Recursion, is a function that calls itself based on a terminating condition, makes use of the stack. Using LIFO, a call to a recursive function saves the return address so that it knows how to return to the calling function after the call terminates.

8. What is merge sort?

Answer:

Merge sort is an efficient and very popular sorting algorithm. It basically follows the strategy that includes divide, conquer and combine. It divides the list recursively into two halves until it can no more be divided. And Merge the smaller lists into new list in sorted order.



9. What is the primary advantage of a linked list?

Answer:

Linked list is a dynamic data structure so it can grow and shrink at runtime by allocating and deallocating memory. So there is no need to give initial size of linked list. As size of linked list can increase or decrease at run time so there is no memory wastage.

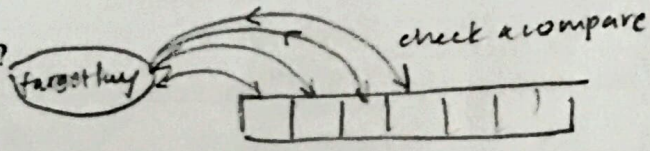
Data structures such as stack and queues can be easily implemented using linked list.

10. What is the difference between a PUSH and a POP?

Answer:

PUSH basically is an operation to insert data into stack and POP is an operation to remove and read data from stack.

11. What is a linear search?



Answer:

In linear search method, each element in the list is checked and compared against the target key. The process is repeated until found or if the end of the file has been reached. Time taken to search elements keeps increasing as the number of elements is increased. Time taken ∝ elements

12. What is the advantage of the heap over a stack?

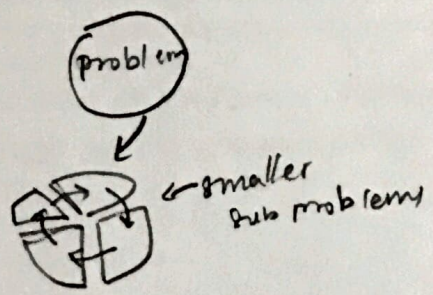
Answer:

The heap is more flexible than the stack. That is because memory space for the heap can be dynamically allocated and de-allocated as needed. However, the memory of the heap can at times be slower when compared to that stack.

13. What is recursive algorithm?

Answer:

Recursive algorithm targets a problem by dividing it into smaller, manageable sub-problems. The output of one recursion after processing one sub-problem becomes the input to the next recursive process.



14. What is binary search?

Answer:

A binary search works only on sorted lists or arrays. A binary search is a quick and efficient method of finding a specific target value from a set of ordered items.

It first checks the middle item and based on that comparison it discards the other half the records. The same procedure is then applied to the remaining half until a match is found or there are no more items left.

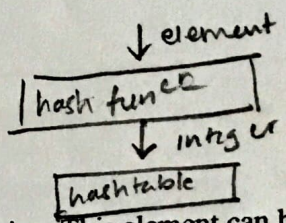
15. What is hashing?

Answer:

Hashing is a technique that is used to uniquely identify a specific object from a group of similar objects.

For example, each student in a college is assigned a unique roll number that can be used to retrieve information about them.

Here, the students are hashed to a unique number.



Hashing is implemented in two steps:

An element is converted into an integer by using a hash function. This element can be used as an index to store the original element, which falls into the hash table.

The element is stored in the hash table where it can be quickly retrieved using hashed key.

Javascript

1. What is JavaScript?

JavaScript is a scripting language with object-oriented capabilities that enables you to create interactive web pages. It is a client side scripting language that executes in the user's browser.

JavaScript can trap user-initiated events such as button clicks, screen touch, dragging screen etc. and can be used for various activities like validating data, displaying popup messages, handling different events of DOM elements etc.

JavaScript does not require expensive development tools; you just need a text editor to write JavaScript code and a browser to display your web page.

2. What is the difference between Java and JavaScript?

JavaScript is very different from language called Java.

The difference between Java and JavaScript are as follows:

Java is an object oriented programming language in which source code is compiled into bytecode and run on a virtual machine. JavaScript is an object based scripting language where source code is interpreted directly by a browser.

Java program has file extension as ".Java". JavaScript file has file extension as ".js".

Java applications can run in any virtual machine(JVM) or browser. JavaScript code runs on browser only.

You require entire Java Development Kit(JDK) to write a Java Program. You just require a text editor to write a JavaScript program.

3. What are the features of JavaScript?

Features of JavaScript

- JavaScript is a lightweight, object-based scripting language.
- JavaScript offers user more control over the browser.
- JavaScript is a client-side technology. It can perform basic calculations on the browser that helps saving server load and network traffic.
- JavaScript validates user input for errors before sending the data over to the server.
- JavaScript is a cross-platform scripting language. All browsers can interpret JavaScript, it solves the problem of compilation and compatibility.
- JavaScript can also dynamically generate HTML content for the web.
- JavaScript can detect the User's Browser and OS information.
- JavaScript is case sensitive.

4. What are the advantages of JavaScript?

Following are the advantages of using JavaScript:

- JavaScript is a simple language to learn and implement.
- JavaScript is a client side language. You can validate user input, perform basic calculations etc. on the browser that helps saving server load and network traffic.
- JavaScript requires no compilation process. The browser interprets JavaScript as HTML tags.
- JavaScript is platform independent language. All modern browsers can understand and interpreted JavaScript code.

- JavaScript helps to provide rich interface to the websites such as Drag and drop components, sliders etc.
- JavaScript provides all capabilities of a procedural language.

5. What are the scopes of a variable in JavaScript?

Scope refers to the visibility of variables. JavaScript has two scopes - global and local.

Global Variables - A global variable has global scope which means it can be accessible from anywhere in your code.

Local Variables - A local variable is one which is declared within a JavaScript function and it remains visible only within that function.

6. What is the purpose of 'This' operator in JavaScript?

'This' keyword refers to the object from where it was called.

7. What is the difference between the operators '==' & '===''?

== operator makes type correction and automatically converts one type into another and return value based upon content equality.

=== operator supports strict equality and only return true if both variables are of same type and also contain same value.

9. What do mean by NULL in Javascript?

'NULL' implies no value or no object. It can be assigned to a variable as a representation of no value.

10. What is the difference between undeclared and undefined?

Undeclared variables are those that are not initialized or declared. If you try to get the value of an undeclared variable, then a runtime error is encountered.

Undefined variables are those that are declared in the program but have not been assigned with any value. If you try to get the value of an undefined variable, an undefined value is returned.

11. What is the difference between innerHTML and innerText?

innerHTML- It interprets an HTML tag if found in a string as HTML and sets the same content in HTML format.

innerText - It interprets an HTML tag as text and sets the content of the tag as plain text.

12. What is an event bubbling in JavaScript?

Event Bubbling is a technique in which the event is first captured and handled by the target element and then propagated to parent elements until it reaches the document object.

Example

```
<body onclick="alert('body')">
 <div onclick="alert('div')">
  <span onclick="alert('span')">span</span>
 </div>
</body>
```

In the above example, when a click event happens on , it first runs the handlers on followed by on the outer and then on the outer <body>.

13. What is NaN in JavaScript?

NaN is a special value that denotes Not-a-Number and cannot be represented as a meaningful number. There are many ways in which NaN can happen such as dividing zero by zero, converting a non-numeric string into a number etc.

We can use isNan() function to see if a value is an NaN value. The function returns true if the argument passed is not a number otherwise it returns false.

14. What is break and continue statements?

Break statement is used to jump out of current loop without executing loop statement(s) below it.

Continue statement skips the rest of the loop statement(s) and continues with the next iteration in the loop.

15. What is the use of typeof operator?

typeof is an operator that returns string description of the operand indicating it's data type.

Example

```
typeof "TutorialRide.com"       // Returns "string"
typeof 123              // Returns "number"
```

16. What are the different types of errors in JavaScript?

There are three types of errors:

Syntax Errors: These errors are errors in the syntax of the source code of a program. They occur at interpret time in JavaScript.

Run-time errors: These errors occur during execution while the program is running. For example, division of a number by zero, calling invalid functions etc.

Logical error: There errors occur when you make a mistake in the logic. Your program compiles and runs without error but doesn't produce intended result. They are hard to determine and debug as they tend to be hidden in the source code.

17. What is the role of a strict mode in JavaScript?

JavaScript is 'not very strict' in throwing errors. Sometimes, it displays result even when there are some errors. To overcome this problem we can use JavaScript strict mode. In 'Strict mode' JavaScript code throws all types of errors. This reduces bugs and makes debugging easier and thus, helps developers to avoid unnecessary mistakes.

To enable strict mode, write "use strict" at the top of the script.

18. Can you access Cookie using javascript?

Yes, we can access cookie using javascript. Cookies can be created, read and erased by JavaScript. They are accessible through the property document.cookie.

19. What is the difference between 'null' and 'undefined'?

Both the keywords signify absence of value.

The differences are:

'null' is a special value represents 'no value'. 'null' can be assigned to a variable that signifies that we have defined a variable but have not assigned any value yet.

'undefined' means a variable has been declared but not defined yet. JavaScript assigns 'undefined' to any object that has been declared but not initialized or defined.