C,C++, COP , DS, DBT , WPT , Core and Advanced Java , SDM , OS , MS.NET

C Programming

1. What is meant by null pointer and void pointer?

A null pointer in C is a pointer that is assigned to zero or NULL where a variable that has no valid address. The null pointer usually does not point to anything. A null pointer can be associated with any data type - int, char, void, etc.

A void pointer in C is a pointer that does not have any associated data type. A void pointer in C clearly indicates that it is empty and can only capable of holding the addresses of any type. Void pointers with addresses can be typecast into any other type easily.

2. Differentiate between Actual Parameters and Formal Parameters.

The Parameters which are sent from main function to the subdivided function are called as **Actual Parameters** and the parameters which are declared a the Subdivided function end are called as **Formal Parameters.**

3. What is the difference between declaring a header file with < > and " "?

If the Header File is declared using < > then the compiler searches for the header file within the Built-in Path. If the Header File is declared using " " then the compiler will search for the Header File in the current working directory and if not found then it searches for the file in other locations.

4. Which statement is efficient and why? x=x+1; or x++; ?

x++; is the most efficient statement as it just a single instruction to the compiler while the other is not. x=x+1; will take more time for execution as it need to perform more read / write operations than x++;

5. What is scope, visibility and lifetime of variable in C?

- **Scope** is defined as the availability of a variable inside a program, scope is basically the region of code in which a variable is available to use.
  There are four types of scope:
  - file scope,
  - block scope,
  - function scope and
  - prototype scope.
- **Visibility** of a variable is defined as if a variable is accessible or not inside a particular region of code or the whole program.
  For e.g.

```
#include <stdio.h>

int main() {
```

```
        int scope; // outer scope variable
        scope = 10;
        // inner block of code
        {
                float scope; // inner scope
                scope = 2.98;
                printf("Inner block scope : %f\n", scope);
        }
        printf("Outer block scope : %d\n", scope);
        return 0;
  }
```

Output :

Inner block scope : 2.980000

Outer block scope : 10

- **Lifetime of a variable** is the time for which the variable is taking up a valid space in the system's memory, it is of three types:
  - static lifetime,
  - automatic lifetime and
  - dynamic lifetime.

C++ Programming

1. Define an Inline Function in C++? Write its syntax. Is it possible for the C++ compiler to ignore inlining?

In order to reduce the function call overhead, C++ offers inline functions. As the name suggests, an inline function is expanded in line when it is called.

As soon as the inline function is called, the whole code of the same gets either inserted or substituted at the particular point of the inline function call. The substitution is complete by the C++ compiler at compile time. Small inline functions might increase program efficiency.

The syntax of a typical inline function is:

```
Inline return-type function-name(parameters)
{
// Function code goes here
}
```

As the inlining is a request, not a command, the compiler can ignore it.

2. Explain Virtual Functions and the concept of Runtime Polymorphism in C++ with a code example.

Any function starting with the virtual keyword is a virtual function. Unlike normal functions that are called in accordance with the type of pointer or reference used, virtual functions are called as per the type of the object pointed or referred.

In other words, virtual functions resolve at runtime and not at compile time. Use of virtual functions focuses the concept of runtime polymorphism. Things required for writing a virtual function in C++ are:

- A base class
- A derived class
- A function with the same name in both the classes i.e. the base class and the derived class
- A pointer or reference of base class type that points or refers, respectively to an object of the derived class

3. Explain the significance of vTable and vptr in C++ and how the compiler deals with them

vTable is a table containing function pointers. Every class has a vTable. vptr is a pointer to vTable. Each object has a vptr. In order to maintain and use vptr and vTable, the C++ compiler adds additional code at two places:

1. In every constructor – This code sets vptr:
    1. Of the object being created
    2. To point to vTable of the class
2. Code with the polymorphic functional call – At every location where a polymorphic call is made, the compiler inserts code in order to first look for vptr using the base class pointer or reference. The vTable of a derived class can be accessed once the vptr is successfully fetched.

4. Define the Reference variable?

The reference variable in C++ is the name (alias) given to the existing variables. The variable name and reference variable point share the same memory location in C++, which helps in updating the original variable using the reference variable.

```
#include<iostream>
using namespace std;
int main()
{
 int x = 10;
 // ref is a reference to x.
 int& ref = x;
 // Value of x is now changed to 20
 ref = 20;
 cout << "x = " << x << endl ;
 // Value of x is now changed to 30
 x = 30;
 cout << "ref = " << ref << endl ;
 return 0;
}
```

5. What is a destructor? Can we overload a destructor?

A destructor is the member function of the class. It has the same name as the class name and also prefixed with a tilde symbol. It can be executed automatically whenever an object loses its scope.

A destructor cannot be overloaded, and it has the only form without the parameters.

Concept of Programming (COP)

1. Why are pointers not used in Java?

Java doesn't use pointers because they are unsafe and increase the complications of the program. Since JVM is responsible for implied memory allocation, thus to avoid direct access to memory by the user, pointers are not allowed in Java.

2. What is the use of System class in Java?

Java System class is one of the core classes. One of the easiest ways to log information for debugging is System.out.print() method. System class is final so we can't subclass and override its behavior through inheritance.

System class doesn't provide any public constructors, so we can't instantiate this class and that's why all of its methods are static. Some of the utility methods of System class are for array copy, get the current time, and reading environment variables.

3. What do you understand by an instance variable and a local variable?

Instance variables are those variables that are accessible by all the methods in the class. They are declared outside the methods and inside the class. These variables describe the properties of an object.

All the objects of the class will have their copy of the variables for utilization. If any modification is done on these variables, then only that instance will be impacted by it, and all other class instances continue to remain unaffected.

Local variables are those variables present within a block, function, or constructor and can be accessed only inside them. The utilization of the variable is restricted to the block scope. Whenever a local variable is declared inside a method, the other class methods don't have any knowledge about the local variable.

4. What are the default values assigned to variables and instances in java?

There are no default values assigned to the variables in java. We need to initialize the value before using it. Otherwise, it will throw a compilation error of (Variable might not be initialized).

But for instance, if we create the object, then the default value will be initialized by the default constructor depending on the data type.

- If it is a reference, then it will be assigned to null.
- If it is numeric, then it will assign to 0.
- If it is a boolean, then it will be assigned to false. etc.

5. Briefly explain the concept of constructor overloading

Constructor overloading is the process of creating multiple constructors in the class consisting of the same name with a difference in the constructor parameters. Depending upon the number of parameters and their corresponding types, distinguishing of the different types of constructors is done by the compiler.

Operating System

1. What is RR scheduling algorithm?

RR (round-robin) scheduling algorithm is primarily aimed for time-sharing systems. A circular queue is a setup in such a way that the CPU scheduler goes around that queue, allocating CPU to each process for a time interval of up to around 10 to 100 milliseconds.

2. What is kernel?

A kernel is the central component of an operating system that manages the operations of computers and hardware. It basically manages operations of memory and CPU time. It is a core component of an operating system. Kernel acts as a bridge between applications and data processing performed at the hardware level using inter-process communication and system calls.

3. What are types of OS and give one example of each type?

1. Batched OS (Example: Payroll System, Transactions Process, etc.)
2. Multi-Programmed OS (Example: Windows O/S, UNIX O/S, etc.)
3. Timesharing OS (Example: Multics, etc.)
4. Distributed OS (LOCUS, etc.)
5. Real-Time OS (PSOS, VRTX, etc.)

-

4. What are different types of variables mostly used in shell scripting?

Shell scripts usually have two types of variables:

- System-defined variables: Also called environment variables, these are special built-in variables in the Linux kernel for each shell. They are normally defined in capital letters by the OS (Linux) and are standard variables.
  Example:
  SHELL
  It is a Unix Defined or System Variable, which specifies the default working shell.
- User-defined variables: These variables are created and defined by users in order to store, access, read, and manipulate data. In general, they are defined in lowercase letters. The Echo command allows you to view them.
  Example:
  $ a=10
  In this case, the user has defined the variable ‘a’ and assigned it the value 10.
-

5. What do you mean by meta-characters in shell script? List them.

In a data field or program, meta-characters are special characters that provide information about other characters. In shells, they're called regular expressions. A character that is neither a letter nor a number is generally considered a meta-character. Using shell meta-characters, you can group together commands, redirect and pipe input/output, put commands in the background, reduce the size of file names and path names, etc. You can use them as wildcards to specify a file name without typing in the file's full name. The most common meta-characters are as follows:

- Asterisk (*): Use the * to match any character.
- Question Mark (?): It matches a single character in the filename.
- Brackets ([…]): used here match some specified characters.
- Hyphen (-): When placed within [] (brackets), the hyphen meta-character matches a specified range.

Core Java

1. What is the final keyword in Java?

The final keyword is the special word in Java; it is used for non- access adjective. The final keyword can be used in a different context, such as:

- **Final Method**- When the method is declared final and can't reverse a decision by the inheriting class
- **Final Class**- When a class is declared as final and can be extended by any subclass class, but it can reach other class
- **Final Variable**- When the final keyword is used with the variable then the value can't be changed once assigned if in case no cost is there to the final variable then using only the class assembler a value can be attributed to it.

2. What is the use of Classloader in Java?

A Java program is made up of a different number of custom classes and pre-defined classes. When a program is executed, JVM is used to load all the content of that needed class and through the use of Classloader JVM, it finds that class.

There are three types of Classloaders:

- **System Class Loader :** It loads all the classes from the classpath.
- **Extension ClassLoader :** It loads all the classes from the extension directory.
- **Bootstrap Class Loader :** It loads all the pre-defined java classes.

3. What are finally and finalize in Java?

Finally block is used with a try-catch block to put the code that you always want to get executed even the execution is thrown by the try-catch block. Finally is just used for releasing the resources which were created by the try block.

Finalize() is a special method in Object class that we can override in our classes. Finalize() is called by the Garbage collector to collect the garbage value when the object is getting it. This

method is generally overridden to release the system resources when garbage value is collected from the object.

4. What is an Iterator?

Iterator interface provides methods to iterate over any collection. We can get iterator instance from a collection using iterator() method. Iterator takes the place of Enumeration in the Java Collection Framework. The iterator allows the caller to remove elements from the underlying collection during the iteration. Java Collection iterator provides a generic way for transversal elements of a collection and implements Iterator Design Pattern.

5. Which is more preferred to synchronize threads – Synchronized method or Synchronized block?

The synchronized block is more preferred because it doesn't lock the object, synchronized methods lock the object and if there are multiple synchronization blocks in the class, even though they are not related, it will stop the execution and put them in a wait state to get the lock on the object.

Advanced Java

1. What are the differences between forward() method and sendRedirect() methods?

- forward() sends the same request to another resource. Whereas sendRedirect() method sends new request always because it uses the URL bar of the browser.
- forward() method works at server side. sendRedirect() method works at client side.
- forward() method works within the server only. sendRedirect() method works within and outside the server.
-

2. What are the steps to connect to a database in java?

- Registering the driver class
- Creating connection
- Creating statement
- Executing queries
- Closing connection

3. Name the different modules of the Spring framework.

- Spring Context – for dependency injection.
- Spring AOP – for aspect oriented programming.
- Spring DAO – for database operations using DAO pattern
- Spring JDBC – for JDBC and DataSource support.
- Spring ORM – for ORM tools support such as Hibernate
- Spring Web Module – for creating web applications.
- Spring MVC – Model-View-Controller implementation for creating web applications, web services etc.

4. What is autowiring in Spring? What are the autowiring modes?

Autowiring enables the programmer to inject the bean automatically. We don't need to write explicit injection logic. Let's see the code to inject bean using dependency injection.

```
<bean id="emp" class="com.javatpoint.Employee" autowire="byName" />
```

The autowiring modes are given below:

1. no : this is the default mode, it means autowiring is not enabled.
2. byName : Injects the bean based on the property name. It uses setter method.
3. byType : Injects the bean based on the property type. It uses setter method.
4. constructor : It injects the bean using constructor

5. What is Hibernate Framework?

Object-relational mapping or ORM is the programming technique to map application domain model objects to the relational database tables. Hibernate is Java-based ORM tool that provides a framework for mapping application domain objects to the relational database tables and vice versa.

Hibernate provides a reference implementation of Java Persistence API, that makes it a great choice as ORM tool with benefits of loose coupling. We can use the Hibernate persistence API for CRUD operations. Hibernate framework provide option to map plain old java objects to traditional database tables with the use of JPA annotations as well as XML based configuration.

Similarly, hibernate configurations are flexible and can be done from XML configuration file as well as programmatically.

Data Structures

1. What is a stack? What are the applications of stack?

Stack is a linear data structure that follows LIFO (Last In First Out) approach for accessing elements. It has one end open called top end from where all insertions and deletions are made. Push, pop, and top (or peek) are the basic operations of a stack.

Following are some of the applications of a stack:

- Check for balanced parentheses in an expression
- Evaluation of a postfix expression
- Problem of Infix to postfix conversion
- Reverse a string

2. What is a queue? What are the applications of queue?

A queue is a linear data structure that follows the FIFO (First In First Out) approach for accessing elements. It has two ends open, called as front and rear, all insertions are made from one end called rear and all deletions are made from front end.

Dequeue from the queue, enqueue element to the queue, get front element of queue, and get rear element of queue are basic operations that can be performed.

Some of the applications of queue are:

- CPU Task scheduling
- BFS algorithm to find shortest distance between two nodes in a graph.

- Website request processing
- Used as buffers in applications like MP3 media player, CD player, etc.
- Managing an Input stream

3. What are different ways of representing the graph?

We can represent a graph in 2 ways:
1. Adjacency matrix: Used for sequential data representation
2. Adjacency list: Used to represent linked data

4. What is meant by indegree and outdegree of vertex?

In case of directed graph, the number of incoming edges towards the vertex is called indegree of that vertex and number of outgoing edges from the vertex is called outdegree of that vertex.

5. How do you know when to use DFS over BFS?

- The usage of DFS heavily depends on the structure of the search tree/graph and the number and location of solutions needed. Following are the best cases where we can use DFS:
- If it is known that the solution is not far from the root of the tree, a breadth first search (BFS) might be better.
- If the tree is very deep and solutions are rare, depth first search (DFS) might take an extremely long time, but BFS could be faster.
- If the tree is very wide, a BFS might need too much memory, so it might be completely impractical. We go for DFS in such cases.
- If solutions are frequent but located deep in the tree we opt for DFS.

6. What is a priority queue?

- A priority queue is an abstract data type that is like a normal queue but has priority assigned to elements.
- Elements with higher priority are processed before the elements with a lower priority.
- In order to implement this, a minimum of two queues are required - one for the data and the other to store the priority.

7. What is an AVL Tree?

AVL trees are height balancing BST. AVL tree checks the height of left and right sub-trees and assures that the difference is not more than 1. This difference is called Balance Factor and is calculates as. BalanceFactor = height(left subtree) − height(right subtree)

8. What is a heap data structure?

Heap is a special tree-based non-linear data structure in which the tree is a complete binary tree. A binary tree is said to be complete if all levels are completely filled except possibly the last level and the last level has all elements towards as left as possible. Heaps are of two types:

1. Max-Heap:
   In a Max-Heap the data element present at the root node must be greatest among all the data elements present in the tree.

This property should be recursively true for all sub-trees of that binary tree.
2. Min-Heap:
In a Min-Heap the data element present at the root node must be the smallest (or minimum) among all the data elements present in the tree.
This property should be recursively true for all sub-trees of that binary tree.

9. What is a doubly-linked list (DLL)? What are its applications.

This is a complex type of a linked list wherein a node has two references:

One that connects to the next node in the sequence
Another that connects to the previous node.

This structure allows traversal of the data elements in both directions (left to right and vice versa).
Applications of DLL are:

- A music playlist with next song and previous song navigation options.
- The browser cache with BACK-FORWARD visited pages
- The undo and redo functionality on platforms such as word, paint etc, where you can reverse the node to get to the previous page.

10. What are the advantages of Linked List over an array?

- The size of a linked list can be incremented at runtime which is impossible in the case of the array.
- The List is not required to be contiguously present in the main memory, if the contiguous space is not available, the nodes can be stored anywhere in the memory connected through the links.
- The List is dynamically stored in the main memory and grows as per the program demand while the array is statically stored in the main memory, size of which must be declared at compile time.
- The number of elements in the linked list are limited to the available memory space while the number of elements in the array is limited to the size of an array.

11. What is a deque?

Deque (also known as double-ended queue) can be defined as an ordered set of elements in which the insertion and deletion can be performed at both the ends, i.e. front and rear.

Database Technologies

1. What is normalization? What are the different types of Normalization?

Normalization is the process of removing redundant data from the database by splitting the table in a well-defined manner in order to maintain data integrity. This process saves much of the storage space.
Different types of Normalization are:

- First Normal Form (1NF): A relation is said to be in 1NF only when all the entities of the table contain unique or atomic values.

- Second Normal Form (2NF): A relation is said to be in 2NF only if it is in 1NF and all the non-key attribute of the table is fully dependent on the primary key.
- Third Normal Form (3NF): A relation is said to be in 3NF only if it is in 2NF and every non-key attribute of the table is not transitively dependent on the primary key.
- BCNF : is the Boyce Code Normal form. It is the higher version of 3Nf which does not have any multiple overlapping candidate keys.

2. What are the advantages and disadvantages of views in the database?

Advantages of Views:

- As there is no physical location where the data in the view is stored, it generates output without wasting resources.
- Data access is restricted as it does not allow commands like insertion, updation, and deletion.

Disadvantages of Views:

- The view becomes irrelevant if we drop a table related to that view.
- Much memory space is occupied when the view is created for large tables.

3. What is Join?

The Join operation is one of the most useful activities in relational algebra. It is most commonly used way to combine information from two or more relations. A Join is always performed on the basis of the same or related column. Most complex queries of SQL involve JOIN command.
There are following types of join:

- Inner joins: Inner join is of 3 categories. They are:
  - Theta join
  - Natural join
  - Equi join
- Outer joins: Outer join have three types. They are:
  - Left outer join
  - Right outer join
  - Full outer join

4. What are some of the advantages of MongoDB?

- MongoDB supports field, range-based, string pattern matching type queries. for searching the data in the database
- It support primary and secondary index on any fields
- It basically uses JavaScript objects in place of procedures
- MongoDB uses a dynamic database schema
- It is very easy to scale up or down
- It has inbuilt support for data partitioning (Sharding).

5. What is a Document, collection and database in MongoDB?

A **Document** in MongoDB is an ordered set of keys with associated values. It is represented by a map, hash, or dictionary. In JavaScript, documents are represented as objects: {"greeting" : "Hello world!"}

Complex documents will contain multiple key/value pairs:

{"greeting" : "Hello world!", "views" : 3}

A **collection** in MongoDB is a group of documents. If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table. Documents within a single collection can have any number of different "shapes.", i.e. collections have dynamic schemas.

For example, both of the following documents could be stored in a single collection:

{"greeting" : "Hello world!", "views": 3}

{"signoff": "Good bye"}

MongoDB groups collections into **databases**. MongoDB can host several databases, each grouping together collections.

Some reserved database names are as follows:

- admin
- local
- config

Software Development Methodologies

1. What is mean by software scope?

Software scope is a well-defined boundary. It includes all kind of activities that are done to develop and deliver the software product.

The software scope defines all functionalities and artifacts to be delivered as a part of the software. The scope also identifies what the product will do? What is not the part of the project? What is project estimation?

This process is helpful to estimate various aspects of the software product. This estimation can be decided either consulting experts or by using pre-defined formulas.

2. What are the advantages of the incremental model?

- The incremental model can be accepted when there is less number of people include in the project.
- Technical risks can be handle with each increment.
- For a minimal period, at least the core product can be delivered to the user.

3. Can you tell something about docker container?

- Docker containers consist of applications and all their dependencies. They share the kernel and system resources with other containers and run as isolated systems in the host operating system.
- The main aim of docker containers is to get rid of the infrastructure dependency while deploying and running applications. This means that any containerized application can run on any platform irrespective of the infrastructure being used beneath.
- Technically, they are just the runtime instances of docker images.

4. What is manual testing? What are advantages and disadvantages of manual testing?

In manual testing, a tester manually verifies the functionality of the software. The tester has a comprehensive list of all the test cases they should test, along with the test data. They go through each case, one by one. They launch the software as an end-user would, enter the input, and manually verify the output.

Disadvantages : Manual testing is inefficient when compared to automated testing. It is slow, not repeatable in a consistent manner, and prone to human misjudgment.

Advantages : Manual testing allows the tester to realistically test the software, using actual user data in a natural user environment, subject to similar external conditions. Only a human, not a computer, can evaluate the usability and accessibility of the application and how it looks and feels to the end-user. It also gives a broader perspective of the system. Finally, some test scenarios just can't be automated and need to be manually tested.

5. What are cloud delivery models?

Cloud delivery models are models that represent the computing environments. These are as follows:

- **Infrastructure as a Service (IaaS):** Infrastructure as a Service (IaaS) is the delivery of services, including an operating system, storage, networking, and various utility software elements, on a request basis.
- **Platform as a Service (PaaS):** Platform as a Service (PaaS) is a mechanism for combining Infrastructure as a Service with an abstracted set of middleware services, software development, and deployment tools. These allow the organization to have a consistent way to create and deploy applications on a cloud or on-premises environment.
- **Software as a Service (SaaS):** Software as a Service (SaaS) is a business application created and hosted by a provider in a multi-tenant model.
- **Function as a Service (FaaS):** Function as a Service (FaaS) gives a platform for customers to build, manage and run app functionalities without the difficulty of maintaining infrastructure. One can thus achieve a "serverless" architecture.

Web Programming Technologies

1. How to integrate CSS on a web page?

There are three methods to integrate CSS on web pages.

1. Inline method - It is used to insert style sheets in HTML document
2. Embedded/Internal method - It is used to add a unique style to a single document
3. Linked/Imported/External method - It is used when you want to make changes on multiple pages.

2. Difference between " == " and " === " operators of javascript.

Both are comparison operators. The difference between both the operators is that,"==" is used to compare values whereas, " === " is used to compare both value and types.

```
var x = 2;
var y = "2";
(x == y)  // Returns true since the value of both x and y is the same

(x === y) // Returns false since the typeof x is "number" and typeof y is "string"
```

3. What is useState() in React?

The useState() is a built-in React Hook that allows you for having state variables in functional components. It should be used when the DOM has something that is dynamically manipulating/controlling.

4. What are the difference between AJAX and Javascript?

- AJAX sends request to the server and does not wait for the response. It performs other operations on the page during that time. Whereas JavaScript make a request to the server and waits for response.
- AJAX does not require the page to refresh for downloading the whole page. JavaScript manages and controls a Web page after being downloaded.
- AJAX minimizes the overload on the server since the script needs to request once. JavaScript posts a request that updates the script every time.

5. What is Node.js and how it works?

Node.js is a virtual machine that uses JavaScript as its scripting language and runs Chrome's V8 JavaScript engine. Basically, Node.js is based on an event-driven architecture where I/O runs asynchronously making it lightweight and efficient. It is being used in developing desktop applications as well with a popular framework called electron as it provides API to access OS-level features such as file system, network, etc.

Dot Net

1. What is meant by CTS component of dot Net framework?

CTS specifies a standard that will mention which type of data and value can be defined and managed in memory during runtime.

It will make sure that programming data defined in different languages should interact with each other for sharing the information. For example, in VB.NET we define datatype as integer, while in C# we define int as a data type.

It can be used to prevent data loss when you are trying to transfer data from a type in one language to its equivalent type in another language.

2. What is a delegate in .NET?

A delegate is a .NET object which defines a method signature and it can pass a function as a parameter.

Delegate always points to a method that matches its specific signature. Users can encapsulate the reference of a method in a delegate object.
When we pass the delegate object in a program, it will call the referenced method. To create a custom event in a class, we can make use of delegate.

3. What are the different types of cookies in ASP.NET?

- **Session Cookie**: It resides on the client machine for a single session until the user logs out.
- **Persistent Cookie**: Resides on the user machine for a period specified for its expiry. It may be an hour, a month or never.

4. What is IIS?

IIS stands for Internet Information Services. It is a powerful web server developed by Microsoft. IIS can also act as a load balancer to distribute incoming HTTP requests to different application servers to allow high reliability and scalability.
It can also act as a reverse proxy, i.e. accept a client's request, forward it to an application server, and return the client's response. A reverse proxy improves the security, reliability, and performance of your application.
A limitation of IIS is that it only runs on Windows. However, it is very configurable. You can configure it to suit your application's specific needs.

5. What is model binding in ASP.NET?

Controllers and views need to work with data that comes from HTTP requests. For example, routes may provide a key that identifies a record, and posted form fields may provide model properties. The process of converting these string values to .NET objects could be complicated and something that you have to do with each request. Model binding automates and simplifies this process.
The model binding system fetches the data from multiple sources such as form fields, route data, and query strings. It also provides the data to controllers and views in method parameters and properties, converting plain string data to .NET objects and types in the process.