Q.2a. What is Java Runtime Environment? Explain the key differences between C++ and Java.

Q2. A) The Java Runtime Environment (JRE) is what you get when you download Java software. The JRE consists of the Java Virtual Machine (JVM), Java platform core classes, and supporting Java platform libraries. The JRE is the runtime portion of Java software, which is all you need to run it in your Web browser. When you download Java software, you only get what you need - no spyware, and no viruses.

C++	Java
Strongly influenced by C syntax, with Object-Oriented features added.	Strongly influenced by C++/C syntax.
Compatible with C source code, except for a few corner cases.	Provides the Java Native Interface and recently Java Native Access as a way to directly call C/C++ code.
Write once, compile anywhere (WOCA).	Write once, run anywhere / everywhere (WORA / WORE).
Allows procedural programming, functional programming, object-oriented programming, generic programming, and template metaprogramming. Favors a mix of paradigms.	Strongly encourages exclusive use of the object-oriented programming paradigm. Includes support for generic programming and creation of scripting languages.
Runs as native executable machine code for the target instruction set(s).	Runs in a virtual machine.
Provides object types and type names. Allows reflection through RTTI.	Is reflective, allowing metaprogramming and dynamic code generation at runtime.
Has multiple binary compatibility standards (commonly Microsoft (for MSVC compiler) and Itanium/GNU (for virtually all other compilers)).	Has a single, OS- and compiler-independent binary compatibility standard, allowing runtime check of correctness of libraries.

Optional automated bounds checking (e.g., the `at()` method in `vector` and `string`containers).	All operations are required to be Bound-checked by all compliant distributions of Java. HotSpotcan remove bounds checking.
Supports native unsigned arithmetic.	No native support for unsigned arithmetic.
Standardized minimum limits for all numerical types, but the actual sizes are implementation-defined. Standardized types are available through the standard library `<cstdint>`.	Standardized limits and sizes of all primitive types on all platforms.
Pointers, references, and pass-by-value are supported for all types (primitive or user-defined).	All types (primitive types and reference types) are always passed by value.[1]
Memory management can be done explicitly through `new / delete`, by automatic scope-based resource management (RAII), or by smart pointers. Supports deterministic destruction of objects. Garbage collection ABI standardized in C++11, though compilers are not required to implement garbage collection.	Automatic garbage collection. Supports a non-deterministic finalize() method whose use is not recommended.[2]

b. Illustrate the concept of passing command line arguments by a Java Program.

b) Command-Line Arguments

Your Java application can accept any number of arguments from the command line. Command-line arguments allow the user to affect the operation of an application. For example, an application might allow the user to specify verbose mode--that is, specify that the application display a lot of trace information--with the command-line argument `-verbose`.

The user enters command-line arguments when invoking the application. For example, suppose you have a Java application, called `Sort`, that sorts lines in a file, and that the data you want sorted is in a file named `friends.txt`. If you are using Windows 95/NT^{TM}, you would invoke the `Sort` application on your data file like this:

c.Explain the terms: **(4)**

(i) JVM
(ii)Web Browser and its examples

```
C:\> java Sort friends.txt
```

In the Java language, when you invoke an application, the runtime system passes the command-line arguments to the application's `main` method via an array of `String`s. Each `String` in the array contains one of the command line arguments. In the previous example, the command-line arguments passed to the `Sort` application is an array that contains a single string: "`friends.txt`".

The Echo Command-Line Arguments Example

This simple application displays each of its command-line arguments on a line by itself:

```
class Echo {
   public static void main (String[] args) {
      for (int i = 0; i < args.length; i++)
         System.out.println(args[i]);
   }
}
```

Try this: Invoke the Echo application. Here's an example of how to invoke the application using Windows 95/NT:

```
C:\> java Echo Drink Hot Java
Drink
Hot
Java
```

You'll notice that the application displays each word--`Drink`, `Hot`, and `Java`--on a line by itself. This is because the space character separates command-line arguments. If you want **Drink Hot Java** to be interpreted as a single argument, you would join them with double quotes, which the system consumes. On Windows 95/NT, you would run it like this:

```
% java Echo "Drink Hot Java"
Drink Hot Java
```

Conventions for Command-Line Arguments

You should observe several conventions when accepting and processing command-line arguments with a Java application.

Parsing Command-Line Arguments

Many applications that accept command-line arguments allow the user to specify various combinations of arguments in various orders. For example, the UNIX command that prints the contents of a directory -- the `ls` utility program -- accepts optional arguments that determine which file attributes to list and the order in which to list the files. This flexibility, which can greatly help the user, requires the application to *parse* the arguments.

Q.3a. "One object in Java can be assigned as reference to another object of the same type." To explain this concept write a complete Java program and show how a reference object works? (8)

ii. A web browser is what you used to get to YA and post this question in the WRONG section.
A web browser is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. An information resource is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video, or other piece of content. Hyperlinks present in resources enable users to easily navigate their browsers to related resources.

Although browsers are primarily intended to access the World Wide Web, they can also be used to access information provided by Web servers in private networks or files in file systems. Some browsers can be also used to save information resources to file systems.
Other software publishers have extended the functionality of Microsoft's Trident engine. The following browsers are all based on the Trident rendering engine:

* AOL Explorer
* Avant Browser
* Bento Browser (built into Winamp)
* Enigma
* GreenBrowser
* Internet Explorer
* iRider
* Maxthon
* MediaBrowser
* MenuBox
* MSN Explorer
* NeoPlanet
* NetCaptor
* RealPlayer
* SlimBrowser
* Tencent Traveler
* TomeRaider
* TheWorld Browser
* UltraBrowser
* WebbIE

b. What is an instance variable? Explain how an instance variable of a class can have different values for each object of that class?

Q3 B) When a number of objects are created from the same class blueprint, they each have their own distinct copies of *instance variables*. In the case of the `Bicycle` class, the instance variables are `cadence`, `gear`, and `speed`. Each `Bicycle` object has its own values for these variables, stored in different memory locations.

Sometimes, you want to have variables that are common to all objects. This is accomplished with the `static` modifier. Fields that have the `static` modifier in their declaration are called *static fields* or *class variables*. They are associated with the class, rather than with any object. Every instance of the class shares a class variable, which is in one fixed location in memory. Any object can change the value of a class variable, but class variables can also be manipulated without creating an instance of the class.

For example, suppose you want to create a number of `Bicycle` objects and assign each a serial number, beginning with 1 for the first object. This ID number is unique to each object and is therefore an instance variable. At the same time, you need a field to keep track of how many `Bicycle` objects have been created so that you know what ID to assign to the next one. Such a field is not related to any individual object, but to the class as a whole. For this you need a class variable, `numberOfBicycles`, as follows:

```
public class Bicycle {

    private int cadence;
    private int gear;
    private int speed;

    // add an instance variable for the object ID
    private int id;

    // add a class variable for the
    // number of Bicycle objects instantiated
    private static int numberOfBicycles = 0;
        ...
}
```

Class variables are referenced by the class name itself, as in

```
Bicycle.numberOfBicycles
```

This makes it clear that they are class variables.

You can use the `Bicycle` constructor to set the `id` instance variable and increment the `numberOfBicycles` class variable:

```
public class Bicycle {

    private int cadence;
    private int gear;
    private int speed;
    private int id;
    private static int numberOfBicycles = 0;

    public Bicycle(int startCadence, int startSpeed, int startGear){
        gear = startGear;
        cadence = startCadence;
        speed = startSpeed;

        // increment number of Bicycles
        // and assign ID number
        id = ++numberOfBicycles;
    }

    // new method to return the ID instance variable
    public int getID() {
        return id;
    }
        ...
}
```

Class Methods

The Java programming language supports static methods as well as static variables. Static methods, which have the `static` modifier in their declarations, should be invoked with the class name, without the need for creating an instance of the class, as in

```
ClassName.methodName(args)
```

Note: You can also refer to static methods with an object reference like
`instanceName.methodName(args)`
but this is discouraged because it does not make it clear that they are class methods.

A common use for static methods is to access static fields. For example, we could add a static method to the `Bicycle` class to access the `numberOfBicycles` static field:

```
public static int getNumberOfBicycles() {
    return numberOfBicycles;
}
```

Not all combinations of instance and class variables and methods are allowed:

- Instance methods can access instance variables and instance methods directly.
- Instance methods can access class variables and class methods directly.
- Class methods can access class variables and class methods directly.

- Class methods ***cannot*** access instance variables or instance methods directly—they must use an object reference. Also, class methods cannot use the `this` keyword as there is no instance for `this` to refer to

Q.4a. What is encapsulation? Explain, how encapsulation provides modularity and information hiding?

Q 4 a) Under this definition, encapsulation means that the internal representation of an object is generally hidden from view outside of the object's definition. Typically, only the object's own methods can directly inspect or manipulate its fields. Some languages like Smalltalk and Ruby only allow access via object methods, but most others (e.g. C++, C# or Java) offer the programmer a degree of control over what is hidden, typically via keywords like public and private.[4] It should be noted that the ISO C++ standard refers to private and public as "access specifiers" and that they do not "hide any information". Information hiding is accomplished by furnishing a compiled version of the source code that is interfaced via a header file.

Hiding the internals of the object protects its integrity by preventing users from setting the internal data of the component into an invalid or inconsistent state. A benefit of encapsulation is that it can reduce system complexity, and thus increases robustness, by allowing the developer to limit the inter-dependencies between software components.

Almost always, there is a way to override such protection – usually via reflection API (Ruby, Java, C#, etc.), sometimes by mechanism like name mangling (Python), or special keyword usage like friend in C++.

Below is an example in C# that shows how access to a data field can be restricted through the use of a private keyword:

```
namespace Encapsulation
{
    class Program
    {
        public class Account
        {
            private decimal accountBalance = 500.00m;

            public decimal CheckBalance()
            {
                return accountBalance;
            }
        }
        static void Main()
        {
            Account myAccount = new Account();
            decimal myBalance = myAccount.CheckBalance();

            /* This Main method can check the balance via the public
            * "CheckBalance" method provided by the "Account" class
            * but it cannot manipulate the value of "accountBalance" */
        }
    }
}
```

Below is an example in Java:

```
public class Employee {
  private BigDecimal salary = new BigDecimal(50000.00);

  public BigDecimal getSalary() {
    return salary;
  }

  public static void main() {
    Employee e = new Employee();
    BigDecimal sal = e.getSalary();
  }
}
```

b.List the eight basic data types used in JAVA. Give examples. (4)

c.Compare in terms of their functions, the following pairs of statements: (4)

(i) While and do.... while
(ii) While and for
(iii) break and continue

Q.5 a. What is the difference between abstract class and interface? Explain with example how and when to use abstract class and interface?

Q5 A) An abstract class is a special kind of class that cannot be instantiated. So the question is why we need a class that cannot be instantiated? An abstract class is only to be sub-classed (inherited from). In other words, it only allows other classes to inherit from it but cannot be instantiated. The advantage is that it enforces certain hierarchies for all the subclasses. In simple words, it is a kind of contract that forces all the subclasses to carry on the same hierarchies or standards.

What is an Interface?

An interface is not a class. It is an entity that is defined by the word Interface. An interface has no implementation; it only has the signature or in other words, just the definition of the methods without the body. As one of the similarities to Abstract class, it is a contract that is used to define hierarchies for all subclasses or it defines specific set of methods and their arguments. The main difference between them is that a class can implement more than one interface but can only inherit from one abstract class.

Feature	**Interface**	**Abstract class**
Multiple inheritance	A class may inherit several interfaces.	A class may inherit only one abstract class.
Default implementation	An interface cannot provide any code, just the signature.	An abstract class can provide complete, default code and/or just the details that have to be overridden.
Access Modfiers	An interface cannot have access modifiers for the subs, functions, properties etc everything is assumed as public	An abstract class can contain access modifiers for the subs, functions, properties
Core VS Peripheral	Interfaces are used to define the peripheral abilities of a class. In other words both Human and Vehicle can inherit from a IMovable interface.	An abstract class defines the core identity of a class and there it is used for objects of the same type.
Homogeneity	If various implementations only share method signatures then it is better to use Interfaces.	If various implementations are of the same kind and use common behaviour or status then abstract class is better to use.
Speed	Requires more time to find the actual method in the corresponding classes.	Fast
Adding functionality (Versioning)	If we add a new method to an Interface then we have to track down all the implementations of the interface and define implementation for the new method.	If we add a new method to an abstract class then we have the option of providing default implementation and therefore all the existing code might work properly.
Fields and Constants	No fields can be defined in interfaces	An abstract class can have fields and constrants defined

b.Does java support multiple interitance? Justify your answer with an example.

Q.6a. What is an exception? Write an exception subclass which throws an exception if the value of a variable 'age' passed as argument to a method is less than 20

Ans

Q6 A) *Exceptions* are generated when a recognized condition, usually an error condition, occurs during the execution of a method. There are a number of standard error conditions defined in Java, and you may define your own error conditions as well.

When an exception is generated, it is said to be *thrown*.

Java syntax includes a system for managing exceptions, by tracking the potential for each method to throw specific exceptions. Note that:

For each method that could throw an exception, your code must inform the Java compiler that it could throw that specific exception.

The compiler marks that method as potentially throwing that exception, and then requires any code calling the method to handle the possible exception.

There are two ways to handle an exception:

You can try the "risky" code, catch the exception, and do something about it, after which the propagation of the exception ceases.

You can mark the method with the risky statement indcating that it throws an exception. Any statement that calls a method capable of throwing an exception must deal with it.

So, if you use a method in your code that is marked as throwing a particular exception, the compiler will not allow that code unless you handle the exception

Once an exception is thrown, it propagates backward up the chain of methods, from callees to callers, until it is caught. Note that:

If the exception occurs in a try block, the JVM looks to the catch block(s) that follow to see if any of them match the exception type.

The first one that matches will be executed.

If none match, then this method ends, and execution jumps to the method that called this one, at the point where the call was made.

If an exception is not caught in your code (which would happen if main was marked as throwing the exception) then the JVM will catch the exception, end that thread of execution, and print a *stack trace*.

There are cases where the compiler does not enforce these rules. Exceptions that fit this category are called*unchecked exceptions*.

```
public void setAge(int age) throws IllegalArgumentException {
    if (age < 0)
        throw new IllegalArgumentException("Age must be >= 0");
    else
```

```
        this.age = age;
}
```

b.Explain various Stream Classes of Java. Give syntax in Java to read/write characters and bytes.

B) The most basic input and output in Java (System.in and System.out fields that have been used in the Basic I/O) is done using streams. Streams are objects that represent sources and destinations of data. Streams that are sources of data can be read from, and streams that are destinations of data can be written to. A stream in Java is an ordered sequence of bytes of undetermined length. Streams are ordered and in sequence so that the java virtual machine can understand and work upon the stream. Streams are analogous to water streams. They exist as a communication medium, just like electromagnetic waves in communication. The order or sequence of bytes in a Java stream allow the virtual machine to classify it among other streams.

Java has various inbuilt streams implemented as classes in the package java.io like the classes of System.in and System.out. Streams can be classed as both input and output streams. All Java streams are derived from Input Stream (java.io.InputStream) and Output Stream (java.io.OutputStream) classes. They are abstract base classes meant for other stream classes. The System.in is the input stream class derivative and analogically System.out is the output counterpart. Both are basic classes used to directly interact with input and output through console, similarly follows System.err. Also Java has streams to communicate across different parts of a program or even among threads. There are also classes that "filter" streams, changing one format to another (e.g. class DataOutputStream, which translates various primitive types to byte streams).

It is a characteristic of streams that they deal only in one discrete unit of data at a time, and different streams deal with different types of data. If one had a stream that represented a destination for bytes, for example, one would send data to the destination one byte at a time. If a stream was a source of byte data, one would read the data a byte at a time. Because this is the only way to access data from a stream, in this latter case, we wouldn't know when we had read all the data from the stream until we actually got there. When reading a stream, one generally has to check each unit of data each read operation to see if the end of the stream has been reached (with byte streams, the special value is the integer -1, or FFFF hex).

Input streams acquire bytes for our programmed java application/program (e.g. a file, an array, a keyboard or monitor, etc.). InputStream is an abstract class that represents a source of byte data. It has a read() method, which returns the next byte in the stream and a close() method, which should be called by a program when that program is done with the stream. The read() method is overloaded, and can take a byte array to read to. It has a skip() method that can skip a number of bytes, and an available() method that a program can use to determine the number of bytes immediately available to be read, as not all the data is necessarily ready immediately. As an abstract class, it cannot be instantiated, but describes the general behavior of an input stream. A few examples of concrete subclasses would be ByteArrayInputStream, which reads from a byte array, and FileInputStream, which reads byte data from a file.

In the following example, we print "Hello world!" on the screen several times. The number of times the message is printed is stored in a file named source.txt. This file should only contain a integer and should be placed in the same folder of the ConfiguredApplication class.

Q.7 a. Define the syntax of tables, forms and frames of HTML.

Q7 A) Within html, tables provide a convenient mechanism for organizing data within this type of structure. While many refinements are possible, the basic html tags for tables are:

Command	Meaning	Note
<table>	begin a table	Requires corresponding </table>
<tr>	begin a row within the table	Requires corresponding </tr>
<th>	begin a header field within a row of the table	
<td>	begin a data field within a row of the table	

In each case, some additional formatting information may be inserted within each tag. For example,

- `border` may be included within a `<table>` tag to provide a box around each square of the table;
- `align="left"` or `align="center"` or `align="right"` may be included with a `<th>` or `<td>` tag, so the data will be left justified, centered left-to-right, or right-justified within each table box; and
- `valign="top"` may be included with a `<th>` or `<td>` tag, so that the data will begin at the top of a box rather than being centered vertically in each cell.

A simple three-framed page:

```
<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
  <frame src="frame_c.htm">
</frameset>
```

An HTML form with two input fields and one submit button:

```
<form action="demo_form.asp" method="get">
  First name: <input type="text" name="fname"><br>
  Last name: <input type="text" name="lname"><br>
  <input type="submit" value="Submit">
</form>
```

b. Discuss the guidelines to be followed in designing Website and Web page.

Q7 B) Designing a good website isn’t as easy and simple as one might think. To design an exceptional site, there are a few guidelines that should be followed. These tips will not guarantee a successful website but it will provide you with how to develop and maintain admirable web site.

Navigation

To start off, having proper navigation is essential to a good website. Having a straightforward and logical navigation system will help users keep control and enjoy surfing through your site.

If the user keeps getting lost or if the links keep bringing him every which way then theres a problem. If one decides to apply a navigation bar on their website, it is mandatory that it be place at the same location on each webpage.

Generally, the navigation bar is placed across the top or on the left margin. The most frustrating thing for any user is being unable to find what they are searching for when they are positive that your website has what they want.

Your website must ensure that all visitors will be able to locate what their looking for in a quick, easy and straightforward manner without trouble.

Under Construction sites

After users visit your site and see the under construction signs, chances are they wont come back but will instead go visit your competitors.

Details

Secondly, having detail in ones site is crucial. From poor page layout, grammar to spelling errors, these are all point one should be careful with when designing their web site. The smallest mistake in detail can cost you a viewer since it will give your web page a bad image.

Information & Content

Having information, in other words proper content, is also quite important. Building a site with informative, detailed and original content will help your visitors and will also aid your search engine rankings.

If your website is one that sells products or services, having well structured, informative information will definitely be useful. Because content is so important, updating that information regularly is a good idea.

The ability to retrieve correct, valuable and up to date information is what makes the internet so popular. Updates for your website can vary from current world news, new offered products and services to new icon, graphic buttons, page layout and structure. Having an up to date website will show visitors that it is indeed a respectable and fully functional site.

To continue, no matter if your website sells products and services or not, having a visible and easy to find Contact Us link is helpful.

Your website should give a option for all visitors to ask questions or to ask for more information about your website. When designing the Contact Us form or any other form on your site, keep it as short as possible.

People do not enjoy answering a long list of questions. Chances are they will get annoy or bored and check out other websites. So if possible, try and use small forms or in some cases, multiple page forms.

Consistency

Finally, consistency is a key factor in creating a professional website. Having consistency between colors, page layout, structure, icons, fonts, pictures and text will give you page a competent look and will put your visitor in a more comfortable atmosphere to navigate your site.

c. What is meant by server page? Explain.

c) Server Pages are scripts that are embedded in HTML and processed on the server before the page is sent to the client's web browser. This concept is explained generally at server-side scripting.
Examples of server pages implementations are:

- Active Server Pages (ASP)
- Caché Server Pages (CSP)
- Dylan Server Pages (DSP)
- Java Server Pages (JSP)

Webware for Python server pages
The same concept is also used by:

- PHP
- ColdFusion

Q.8 a. Give a XHTML code to create a scroll bar when an element's content is too big to fit.

```
8a) <div style="border:1px solid black;width:150px;height:100px;overflow:scroll;overflow-y:scroll;overflow-x:hidden;">
<p style="height:250%;">
By using overflow-y, we can create scroll bars when the contents of this div are bigger/higher than the container. Also, by setting this paragraph to 250 percent, it is automatically 250 percent higher than the parent container - forcing an overflow.
</p>
</div>
```

b. Give short note on information architecture (IA) and its relevance in web designing.

b) Steps to developing an intuitive website information architecture**:**

- Find out what the mission or purpose of the website is: why will people come to your site?
- Determine the immediate and long-range goals of the site: are they different?
- Pinpoint the intended audiences and conduct a requirements analysis for each group.
- Collect site content and develop a content inventory.
- Determine the website's organizational structure, which can include:

- hierarchical
- narrow and deep
- broad and shallow
- sequential
- tag-based
- Create an outline of the site, which can include:

  Content Inventory: a hierarchical view of the site content, typically in a spreadsheet format, which briefly describes the content that should appear on each page and indicates where pages belong in terms of global and local navigation.

  Site Maps: visual diagrams that reflect site navigation and main content areas. They are usually constructed to look like flowcharts and show how users will navigate from one section to another. Other formats may also indicate the relationships between pages on the site.

Create a visual blueprint of the site, which can include:

Wireframes: rough illustrations of page content and structure, which may also indicate how users will interact with the website. These diagrams get handed off to a visual designer, who will establish page layout and visual design. Wireframes are useful for communicating early design ideas and inform the designer and the client of exactly what information, links, content, promotional space, and navigation will be on every page of the site. Wireframes may illustrate design priorities in cases where various types of information appear to be competing.

Define the navigation systems:

Global navigation: Global navigation is the primary means of navigation through a website. Global navigation links appear on every page of the site, typically as a menu located at the top or side of each web page.

Local navigation: Local links may appear as text links within the content of a page or as a submenu for a section of the website. Local navigation generally appears in the left-hand margin of a web page and sometimes is placed below the global navigation.

Utility links: Utility links appear in the header or footer of every page. These may include infrequently used links such as: Contact Us, About Us, Customer Support, Customer Feedback, Privacy Policy, Terms of Use, Site Map, Press Room, etc. Search boxes often appear in the header of the site as well, so the Search feature is available on every page of the site.

Conduct user research: Once you have a draft navigation structure, conduct appropriate usability research to collect feedback from the target audience. Methods may include: Card Sorting, Cognitive Walkthroughs,Contextual Task Analyses, and Usability Testing,

Q.9 a. Explain CGI architecture. In what situation can CGI be faster than Servlet? Explain.

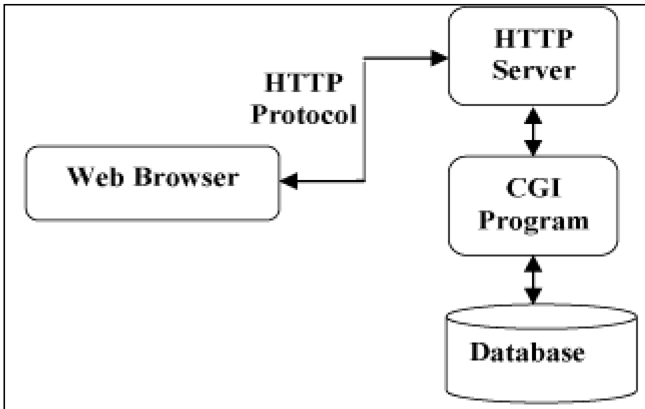
Q9 A)

To understand the concept of CGI, lets see what happens when we click a hyper link to browse a particular web page or URL.

- Your browser contacts the HTTP web server and demand for the URL ie. filename.
- Web Server will parse the URL and will look for the filename in if it finds that file then sends back to the browser otherwise sends an error message indicating that you have requested a wrong file.
- Web browser takes response from web server and displays either the received file or error message.

However, it is possible to set up the HTTP server so that whenever a file in a certain directory is requested that file is not sent back; instead it is executed as a program, and whatever that program outputs is sent back for your browser to display. This function is called the Common Gateway Interface or CGI and the programs are called CGI scripts. These CGI programs can be a PERL Script, Shell Script, C or C++ program etc.

CGI Architecture Diagram



Web Server Support & Configuration

Before you proceed with CGI Programming, make sure that your Web Server supports CGI and it is configured to handle CGI Programs. All the CGI Programs be executed by the HTTP server are kept in a pre-configured directory. This directory is called CGI Directory and by convention it is named as /cgi-bin. By convention PERL CGI files will have extention as **.cgi**.

a CGI will be faster than a servlet when the time to execute the Java code of the servlet takes long enough that the time to load and execute the CGI program is dominated by the time the Java program runs. So, for example, if you had a CGI program in C that had

```
main(){ return; }
```

and you compared that with a servlet, it might well be faster.

What you want to do is set up a servlet container and a CGI directory and actually monitor and measure some comparative performance.

b. Explain the concept of Cascading style sheets in XHTML. Give illustrative examples.

Q 9 B) CSS can be applied to HTML or XHTML using three methods: linked, embedded, and inline. In the linked method, the CSS is stored in a separate file, instead of directly in the HTML page. In the embedded method, CSS is stored as part of the HTML page, in the header section. In the inline method, CSS is stored directly in the style attributes of the HTML tags, as <div style="font-weight:bold">Bold Font</div>.

The neatest method is probably the linked one, but the other ones are convenient and quick in the phases of prototyping a web page. The embedded and inline methods do not require having a separate file. The inline method saves you the trouble of considering what CSS classes your document should have. For a larger site, in which many web pages share the same styling, and in which the styling should be customizable by the user, the linked method is the only viable option.

With linked CSS, CSS rules are stored in a separate file. To refer to that file from the HTML page, add the link element (and its closing element within XHTML) to the head element, as shown in the following example, which assumes that the stylesheet is stored in the file named "style.css".

```
<head>
  <title>Example Web Page</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
```

The link element in the example has three attributes. The first, rel, tells the browser the type of the target of the link. The second, type, tells the browser what type of stylesheet it is. And the third, href, tells the browser under which URL to find the stylesheet. In the example, the URL is relative, but it can also be absolute.

The "style.css" with a single rule contains only the following text:

```
 p {
   font-weight:bold;
 }
```

This tells the browser that the text in the paragraph (p) element should be rendered as bold.

Example rendering:

Text that will be formatted.

The source code for the complete HTML document is thus as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
 "http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
  <head>
    <title>Example Web Page</title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <p>Text that will be formatted.</p>
  </body>
</html>
```

Embedding[edit]

Dynamically generated webpages may need to use embedded CSS but this should be kept to a minimum. Even in dynamic pages any CSS that is common to multiple pages should generally be moved to a linked stylesheet.

Embedded CSS is CSS that is located in the header of the HTML document that requires styling. For example we would like the text in this HTML document to appear bold.

```
 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
 <html lang="en">
  <head>
   <title>Example Web Page</title>
  </head>
  <body>
   <p>
    Text that will be formatted.
   </p>
  </body>
 </html>
```

The CSS must be placed in the document's header section:

```
  <head>
```

```
   <title>Example Web Page</title>
   <style type="text/css">
    p {
     font-weight:bold;
    }
   </style>
  </head>
```

The CSS is contained in a style element. Setting the element's type attribute to text/css tells the browser that the enclosed sheetstyle is written in CSS and should be used to format the page. If the attribute type is missing or set to an unrecognized value the CSS will not be applied to the page.

The little bit of CSS in this example tells the browser to make all the text found in any paragraph (p) elements bold. The text on the page would look like this:

Text that will be formatted.

Here is the complete document including the CSS.

```
 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
 <html lang="en">
  <head>
   <title>Example Web Page</title>
   <style type="text/css">
    p {
     font-weight:bold;
    }
   </style>
  </head>
  <body>
   <p>
    Text that will be formatted.
   </p>
  </body>
 </html>
```

Text Book

1. Programming with Java – A Primer, E. Balagurusamy, Third Edition, TMH, 2007

2. An Introduction to Web Design + Programming, Paul S. Wang and Sanda S. Katila, Thomson Course Technology, India Edition, 2008