1/31/2020

Kajal Shukla

Experience

It contain interview questions of various company.

Company Name: LeewayHerz

Question Type: Practical Question

1. Implementation of Stack (push and pop) method using array in Java?

Solution:

package ComapniesQuestion;

public class stack {

static final int MAX=1000;

int top;

int a[]=new int[MAX];

boolean isEmpty(){

return (top<0);

}

stack () {

top=-1;

}

boolean push (int x) {

if (top>= (MAX-1))

{

System.out.println("Stack overflow");

return false;

}

else{

a[++top]=x;

System.out.println(x+"pushed into stack");

return true;

}}

int pop(){

if (top<0)

{

System.out.println("Stack Underflow");

return 0;

}

else{

int x= a[top--];

return x;

}

}

int peek(){

if(top<0){

System.out.println("Stack Underflow");

return 0;

}

else {

int x=a[top];

return 0;

}

}

}

Company Name: Oodels

Question Type: Practical Question & Theoretical Question

Practical Questions

1. Draw a pascal’s triangle

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

Solution:

package pattern.programs;

public class PascalNumericTriangle {

public static void main(String[] args) {

int i,j,row=5;

for ( i = 0; i < row; i++) {

int number=1;

System.out.printf("%" +(row-i)\*2+ "s", "");

for ( j = 0; j <=i; j++) {

System.out.printf("%4d",number);

number=number\*(i-j)/(j+1);

}

System.out.println();

}

}

}

1. Draw a right pascal’s triangle

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

Solution:

package basicpattern;

import java.util.Scanner;

public class Pattern7 {

public static void main(String[] args) {

Scanner scanner=new Scanner(System.in);

System.out.println("Enter the value");

int value=scanner.nextInt();

for (int i = 0; i <= value; i++) {

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

System.out.print(" \*");

}

System.out.println();}

for (int i = 0; i <= value; i++) {

for (int j = value; j >=i; j--) {

System.out.print(" \*");

}

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

System.out.print(" ");

}System.out.println();

}

}

}

1. Draw a diamond star pattern

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

Solution:

Theoretical Questions:

1) What is advantage of Encapsulation?

Answer:

***Encapsulation in Java binds data and code together***. It is defined as the process in which we wrap the data into a single unit. It basically creates a shield and the code cannot be accessed outside the shield or by any code outside the shield. Java Beans class is the example of a fully encapsulated class.

* The [***variables in Java***](https://data-flair.training/blogs/variables-in-java/) or the method of the class are hidden from any other class and cannot be accessed outside the class.
* We can also call it, as Data-hiding.
* The encapsulated class is easy to test.
* Standard IDE’s like Eclipse, NetBeans are providing the facility to generate getter setter methods, so it is very easy to create an encapsulated call
* It can be achieved by declaring the class as private while the methods as public so that the variables can be accessed.

Advantages of Encapsulation in Java

These are benefits of Encapsulation in Java:

* **Data Hiding –** It can provide the programmer to hide the inner classes and the user to give access only to the desired codes. It allows the programmer to not allow the user to know how [variables](https://en.wikipedia.org/wiki/Variable_(computer_science)) and data store.
* **Getter and Setter Methods –**Private member can only be accessed within the same class. An outside class cannot access the data members of that class. If you need to access these variables, you have to use public “getter” and “setter” methods.
* **Flexibility –** With this, we can make the data as read-only or write-only as we require it to be. It also improves the maintainability and flexibility of code.
* **Reusability –** It allows the user to a programmer to use the existing code again and again in an effective way.
* **Testing of the code –** Ease of testing becomes easy. So it is better for Unit testing

2) What is Inheritance?

Answer:

**Inheritance** is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class.

3) What is multi-threading explain with code?

Answer:

 A thread is a light-weight smallest part of a process that can run concurrently with the other parts (other threads) of the same process. Threads are independent because they all have separate path of execution that’s the reason if an exception occurs in one thread, it doesn’t affect the execution of other threads. All threads of a process share the common memory. ***The process of executing multiple threads simultaneously is known as multithreading.***

## Creating a thread in Java

There are two ways to create a thread in Java:   
1) By extending Thread class.  
2) By implementing Runnable interface.

* getName(): It is used for Obtaining a thread’s name
* getPriority(): Obtain a thread’s priority
* isAlive(): Determine if a thread is still running
* join(): Wait for a thread to terminate
* run(): Entry point for the thread
* sleep(): suspend a thread for a period of time
* start(): start a thread by calling its run() method

### **Method 1: Thread creation by extending Thread class**

class MultithreadingDemo extends Thread{

public void run(){

System.out.println("My thread is in running state.");

}

public static void main(String args[]){

MultithreadingDemo obj=new MultithreadingDemo();

obj.start();

}

}

### **Method 2: Thread creation by implementing Runnable Interface**

class MultithreadingDemo implements Runnable{

public void run(){

System.out.println("My thread is in running state.");

}

public static void main(String args[]){

MultithreadingDemo obj=new MultithreadingDemo();

Thread tobj =new Thread(obj);

tobj.start();

}

}

4)Difference between abstraction and interface.

Answer:

Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1) Abstract class can **have abstract and non-abstract** methods. | Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also. |
| 2) Abstract class **doesn't support multiple inheritance**. | Interface **supports multiple inheritance**. |
| 3) Abstract class **can have final, non-final, static and non-static variables**. | Interface has **only static and final variables**. |
| 4) Abstract class **can provide the implementation of interface**. | Interface **can't provide the implementation of abstract class**. |
| 5) The **abstract keyword** is used to declare abstract class. | The **interface keyword** is used to declare interface. |
| 6) An **abstract class** can extend another Java class and implement multiple Java interfaces. | An **interface** can extend another Java interface only. |
| 7) An **abstract class** can be extended using keyword "extends". | An **interface** can be implemented using keyword "implements". |
| 8) A Java **abstract class** can have class members like private, protected, etc. | Members of a Java interface are public by default. |
| 9)**Example:** public abstract class Shape{ public abstract void draw(); } | **Example:** public interface Drawable{ void draw(); } |

Company Name: Vanisb Technologies

Question Type: Theoretical Question

**Theoretical Question:**

1)What is payment gateway method in e-commerce website?

Answer:

**Payment Gateways** are software and servers that transmit Transaction information to Acquiring Banks and responses from Issuing Banks (such as whether a transaction is approved or declined). Essentially, **Payment Gateways** facilitate communication within banks.

2)What is add to chart method in e-commerce website?

Answer:

Add to Cart is a way to create a [temporary list](http://ask.mysapl.org/a.php?qid=120301) of items by adding them to your cart, which will keep track of the items until you leave our website.

You can [export](http://ask.mysapl.org/a.php?qid=120303) items in your cart by saving the list to a file or sending it to an email address. You can also place the items on [hold](http://ask.mysapl.org/a.php?qid=120273) or add them to your [wish list](http://ask.mysapl.org/a.php?qid=213363).

Step 1 : Create a table cart and insert record when user clicks on add to cart option.

Step 2: Create a bean named cart.

Step 3: When user clicks on add to cart set data in java bean and fire insert query to add data to cart table.

Step 4: If you want to show items added in cart,fire a query and retrieve data mapped to that user in cart table and show cart count.

Step 5: Once user clicks on cart count.Display all the items in detail by retrieving data from cart table.

Company Name: Raw Cube

Question Type: Practical Question & Theoretical Question

**Practical Question:**

1) Write a program to swap Two number without using third variable?

Solution:

package Number;

public class SwappingNumberWithoutThirdVariable {

public static void main(String[] args) {

int a=20,b=30;

System.out.println("Number Before Swap");

System.out.println("First Number : "+a);

System.out.println("Second Number : "+b);

System.out.println("Number After Swap");

a=a+b;

b=a-b;

a=a-b;

System.out.println("First Number : "+a);

System.out.println("Second Number : "+b);

}

}

**Theoretical Question:**

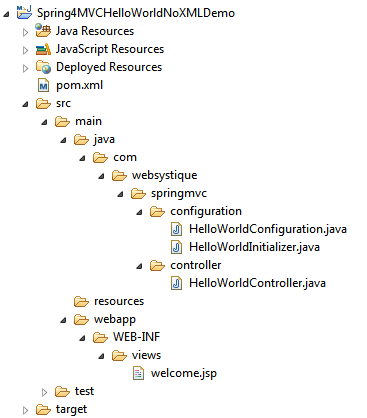
1) What is spring configuration process?

Answer:

#### **Step 1: Create a project with required directory structure**

Post [Creating a maven web project with eclipse](http://websystique.com/maven/create-a-maven-web-project-with-eclipse/) contains step-by-step instruction to create a maven project with eclipse.

Following will be the final Project structure.



#### **Step 2: Update pom.xml with Spring and Servlet dependency**

The Spring java-based configuration we are going to discuss depends on Servlet 3.0 api, so we need to include that as a dependency in pom.xml

#### **Step 3: Add Controller**

Add a controller class under src/main/java

#### **Step 4: Add View**

Create a new folder named views under WEB-INF and add in a Simple JSP page welcome.jsp (WEB-INF/views/welcome.jsp) to simply access the model value sent from controller.

#### **Step 5: Add Configuration Class**

Add the below mentioned class under src/main/java with specified package as shown below. This configuration class can be treated as a replacement of spring-servlet.xml as it contains all the information required for component-scanning and view resolver.

#### **Step 6: Add Initialization class**

Add an initializer class implementing WebApplicationInitializer under src/main/java with specified package as shown below(which in this case acts as replacement of any spring configuration defined in web.xml). During Servlet 3.0 Container startup, this class will be loaded and instantiated and it’s on Startup method will be called by servlet container.

#### **Step 7: Build and Deploy the application**

One thing to keep in mind that the spring java based configuration API’s like WebApplicationInitializer depends on Servlet 3.0 containers. So make sure you don’t have any web.xml with servlet declaration less than 3.0. For our case, we have removed web.xml file from our application.

2) What is Encapsulation?

Answer:

***Encapsulation in Java binds data and code together***. It is defined as the process in which we wrap the data into a single unit. It basically creates a shield and the code cannot be accessed outside the shield or by any code outside the shield. Java Beans class is the example of a fully encapsulated class.

3) Difference between abstraction and interface?

Answer:

Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1) Abstract class can **have abstract and non-abstract** methods. | Interface can have **only abstract** methods. Since Java 8, it can have **default and static methods** also. |
| 2) Abstract class **doesn't support multiple inheritance**. | Interface **supports multiple inheritance**. |
| 3) Abstract class **can have final, non-final, static and non-static variables**. | Interface has **only static and final variables**. |
| 4) Abstract class **can provide the implementation of interface**. | Interface **can't provide the implementation of abstract class**. |
| 5) The **abstract keyword** is used to declare abstract class. | The **interface keyword** is used to declare interface. |
| 6) An **abstract class** can extend another Java class and implement multiple Java interfaces. | An **interface** can extend another Java interface only. |
| 7) An **abstract class** can be extended using keyword "extends". | An **interface** can be implemented using keyword "implements". |
| 8) A Java **abstract class** can have class members like private, protected, etc. | Members of a Java interface are public by default. |
| 9)**Example:** public abstract class Shape{ public abstract void draw(); } | **Example:** public interface Draw able{ void draw(); } |

4) Tell me Encapsulation in the room?

Answer:

Any things that exit in the room are Encapsulation because I don’t know about it.

5) Difference between print and println?

Answer:

| **PRINTLN()** | **PRINT()** |
| --- | --- |
| It adds new line after the message gets displayed. | It does not add any new line. |
| It can work without arguments. | This method only and only works with argument, otherwise it is a syntax error. |

5) What is inheritance?

Answer:

**Inheritance** is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class.

Institute Name: JLJ Group

Question Type: Practical Question & Theoretical Question

**Practical Question:**

1) Write a program to find largest number in three number.

Solution:

2) Write a program to print table using while loop.

Solution:

package Number;

import java.util.Scanner;

public class Table {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int num;

System.out.println("Enter The Value");

num = sc.nextInt();

int i=1;

while(i<=10){

System.out.println(num+ " X "+i+" = "+num\*i);

i++;

}

}

}

3) Write a string program.

Solution:

4) Write a program to find all arm strong number between 0 to 9.

Solution:

Theoretical Question:

1) What is use of scanner class?

Answer:

Scanner is a class in java.util package used for obtaining the input of the primitive types like int, double, etc. and strings. It is the easiest way to read input in a Java program,

2) What is use of util package?

Answer:

Java.util package contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes. This reference will take you through simple and practical methods available in java.util package.

3) What is use of nextInt method?

Answer:

Statement **n = s.nextInt ();** is used to input the value of an integer variable 'n' from the user.

4) What is Object Class?

Answer:

The **Object class** is the parent class of all the classes in java by default. In other words, it is the topmost class of java.

The Object class is beneficial if you want to refer any object whose type you don't know. Notice that parent class reference variable can refer the child class object, known as upcasting.

5) Why we use class in java?

Answer:

Basically class’s help you take all the properties and behaviors of an object in your program, and combine them into a single interface, then re-use that interface wherever you need that type of object in your program.

Company Name: IRB

Question Type: Practical Question

**Practical Question**

1) Write a program to find out duplicate from array.

Solution:

2) Write a program to sort an array using bubble sort.

Solution:

3) Write a program to draw a pattern

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Solution:

package basicpattern;

import java.util.Scanner;

public class Pattern15 {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

System.out.println("Enter The value");

int value=s.nextInt();

for (int i = 0; i <=value; i++) {

for (int j = 0; j <=i ; j++) {

System.out.print("\*");

}System.out.println();

}

}

}

4) Write a program to reverse a value in number.

Solution:

package Number;

import java.util.Scanner;

public class ReversedNumber {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int num,reverse = 0;

System.out.println("Enter the number");

num=sc.nextInt();

System.out.println("Original Number Is : "+num);

while(num!=0){

reverse = reverse\*10;

reverse=reverse+num%10;

num=num/10;

}

System.out.println("Reverse Number is : "+reverse);

}

}

Company Name: OH! Puhleeez Digital marketing solution

Question Type: Practical Question and Theoretical Question

**Practical Question**

1) Write a program to get product information and it can add product into add to cart.

Solution:

**Theoretical Question**

1) What is box model?

Answer:

CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of web pages. It can be used as a toolkit for customizing the layout of different elements. The web browser renders every element as a rectangular box according to the CSS box model.  
Box-Model has multiple properties in CSS. Some of them are given below:

* borders
* margins
* padding
* Content

The following figure illustrates the box model.  


**Border Area:** It is the area between the box’s padding and margin. Its dimensions are given by the width and height of border.

**Margin Area:** This area consists of space between border and margin. The dimensions of Margin area are the margin-box width and the margin-box height. It is useful to separate the element from its neighbors.

**Padding Area:** It includes the element’s padding. This area is actually the space around the content area and within the border box. Its dimensions are given by the width of the padding-box and the height of the padding-box.

**Content Area:** This area consists of content like text, image, or other media content. It is bounded by the content edge and its dimensions are given by content box width and height.

2) What is Meta tag and why it is use?

Answer:

The META elements can be used to include name/value pairs describing properties of the HTML document, such as author, expiry date, a list of keywords, document author etc.

The **<meta>** tag is used to provide such additional information. This tag is an empty element and so does not have a closing tag but it carries information within its attributes.

You can include one or more meta tags in your document based on what information you want to keep in your document but in general, meta tags do not impact physical appearance of the document so from appearance point of view, it does not matter if you include them or not.

You can add metadata to your web pages by placing <meta> tags inside the header of the document which is represented by **<head>** and **</head>** tags. A meta tag can have following attributes in addition to core attributes −

|  |  |
| --- | --- |
| **Sr.No** | **Attribute & Description** |
| 1 | **Name**  Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc. |
| 2 | **Content**  Specifies the property's value. |
| 3 | **Scheme**  Specifies a scheme to interpret the property's value (as declared in the content attribute). |
| 4 | **http-equiv**  Used for http response message headers. For example, http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie. |

3) What you understand from responsive?

Answer:

A responsive [website](https://www.webopedia.com/TERM/W/web_site.html) is one that has been designed to respond, or adapt, based on the technology and type of computing device used by the visitor to display the site. It is basically one website design that will looks good at any size — from a large desktop LCD monitor to the smaller screens we use on smartphones and tablets. Responsive design ensures visitors to the site have a similar experience that is independent of the of the size of the device device used to view the site.

***Three Main Components of Responsive Design***

***Responsive website design consists of the following three main components:***

* ***Flexible layouts*** - Using a flexible grid to create the website layout that will dynamically resize to any width.
* ***Media queries*** - An extension to media types when targeting and including styles. Media queries allow designers to specify different styles for specific browser and device circumstances.
* ***Flexible media*** - Makes media (images, video and other formats) scalable, by changing the size of the media as the size of the viewport changes.

4) Difference between html and html5?

Answer:

|  |  |
| --- | --- |
| **HTML** | **HTML5** |
| It didn’t support audio and video without the use of flash player support. | It supports audio and video controls with the use of <audio> and <video> tags. |
| It uses cookies to store temporary data. | It uses SQL databases and application cache to store offline data. |
| Does not allow JavaScript to run in browser. | Allows JavaScript to run in background. This is possible due to JS Web worker API in HTML5. |
| Vector graphics is possible in HTML with the help of various technologies such as VML, Silver-light, Flash, etc. | Vector graphics is additionally an integral a part of HTML5 like SVG and canvas. |
| It does not allow drag and drop effects. | It allows drag and drop effects. |
| Not possible to draw shapes like circle, rectangle, triangle etc. | HTML5 allows to draw shapes like circle, rectangle, triangle etc. |
| It works with all old browsers. | It supported by all new browser like Firefox, Mozilla, Chrome, Safari, etc. |
| Older version of HTML are less mobile-friendly. | HTML5 language is more mobile-friendly. |
| Doctype declaration is too long and complicated. | Doctype declaration is quite simple and easy. |
| Elements like nav, header were not present. | New element for web structure like nav, header, footer etc. |
| Character encoding is long and complicated. | Character encoding is simple and easy. |
| It is almost impossible to get true Geolocation of user with the help of browser. | One can track the Geolocation of a user easily by using JS Geolocation API. |
| It cannot handle inaccurate syntax. | It is capable of handling inaccurate syntax. |
| Attributes like charset, async and ping are absent in HTML. | Attributes of charset, async and ping are a part of HTML 5. |

There are many HTML elements which have been modified or removed from HTML5. Some of them are listed below:

|  |  |
| --- | --- |
| **ELEMENT** | **IN HTML5** |
| <applet> | Changed to <object> |
| <acronym> | Changed to <abbr> |
| <dir> | Changed to <ul> |
| <frameset> | Removed |
| <frame> | Removed |
| <no frames> | Removed |
| <strike> | No new tag. CSS is used for this |
| <big> | No new tag. CSS is used for this |
| <base font> | No new tag. CSS is used for this |
| <font> | No new tag. CSS is used for this |
| <center> | No new tag. CSS is used for this |
| <tt> | No new tag. CSS is used for this |

5) Describe type of list?

Answer:

HTML lists are used to present list of information in well formed and semantic way. There are three different types of list in HTML and each one has a specific purpose and meaning.

* **Unordered list** — Used to create a list of related items, in no particular order.
* **Ordered list** — Used to create a list of related items, in a specific order.
* **Description list** — Used to create a list of terms and their descriptions.

## HTML Unordered Lists

An unordered list created using the <ul> element, and each list item starts with the <li> element.

The list items in unordered lists are marked with bullets. Here's an example:

#### **Example**

<ul>

<li>Chocolate Cake</li>

<li>Black Forest Cake</li>

<li>Pineapple Cake</li>

</ul>

— The output of the above example will look something like this:

* Chocolate Cake
* Black Forest Cake
* Pineapple Cake

## HTML Ordered Lists

An ordered list created using the <ol> element, and each list item starts with the <li> element. Ordered lists are used when the order of the list's items is important.

The list items in an ordered list are marked with numbers. Here's an example:

#### **Example**

<ol>

<li>Fasten your seatbelt</li>

<li>Starts the car's engine</li>

<li>Look around and go</li>

</ol>

— The output of the above example will look something like this:

1. Fasten your seatbelt
2. Starts the car's engine
3. Look around and go

## HTML Description Lists

A description list is a list of items with a description or definition of each item.

The description list is created using <dl> element. The <dl> element is used in conjunction with the <dt> element which specify a term, and the <dd> element which specify the term's definition.

Browsers usually render the definition lists by placing the terms and definitions in separate lines, where the term's definitions are slightly indented. Here's an example:

#### **Example**

<dl>

<dt>Bread</dt>

<dd>A baked food made of flour.</dd>

<dt>Coffee</dt>

<dd>A drink made from roasted coffee beans.</dd>

</dl>

— The output of the above example will look something like this:

Bread

A baked food made of flour.

Coffee

A drink made from roasted coffee beans.

6) Type of css and also describe when they are used?

Answer:

Cascading Style Sheet(CSS) is used to set the style in web pages which contain HTML elements. It sets the background color, font-size, font-family, color, etc property of elements in a web pages.  
There are three types of CSS which are given below:

* Inline CSS
* Internal or Embedded CSS
* External CSS

**Inline CSS:** Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using style attribute.

**Example:**

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <title>Inline CSS</title>      </head>        <body>          <p style = "color:#009900; font-size:50px;                  font-style:italic; text-align:center;">              GeeksForGeeks          </p>      </body>  </html> |

**Output:**  


**Internal or Embedded CSS:** This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.  
**Example:**

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <title>Internal CSS</title>          <style>              .main {                  text-align:center;              }              .GFG {                  color:#009900;                  font-size:50px;                  font-weight:bold;              }              .geeks {                  font-style:bold;                  font-size:20px;              }          </style>      </head>      <body>          <div class = "main">              <div class ="GFG">GeeksForGeeks</div>                <div class ="geeks">                  A computer science portal for geeks              </div>          </div>      </body>  </html> |

**Output:**  


**External CSS:** External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, … etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using **link** tag. This means that for each element, style can be set only once and that will be applied across web pages.

**Example:** The file given below contains CSS property. This file save with .css extension. For Ex: **geeks.css**

body {

background-color:powderblue;

}

.main {

text-align:center;

}

.GFG {

color:#009900;

font-size:50px;

font-weight:bold;

}

#geeks {

font-style:bold;

font-size:20px;

}

Below is the HTML file that is making use of the created external style sheet

* **link** tag is used to link the external style sheet with the html webpage.
* **href** attribute is used to specify the location of the external style sheet file.

|  |
| --- |
| <!DOCTYPE html>  <html>      <head>          <link rel="stylesheet" href="geeks.css"/>      </head>        <body>          <div class = "main">              <div class ="GFG">GeeksForGeeks</div>              <div id ="geeks">                  A computer science portal for geeks              </div>          </div>      </body>  </html> |

**Output:**  


**Properties of CSS:** Inline CSS has the highest priority, then comes Internal/Embedded followed by External CSS which has the least priority. Multiple style sheets can be defined on one page. If for an HTML tag, styles are defined in multiple style sheets then the below order will be followed.

* As Inline has the highest priority, any styles that are defined in the internal and external style sheets are overridden by Inline styles.
* Internal or Embedded stands second in the priority list and overrides the styles in the external style sheet.
* External style sheets have the least priority. If there are no styles defined either in inline or internal style sheet then external style sheet rules are applied for the HTML tags.

7) What is full form of css?

Answer:

CSS stands for **Cascading Style Sheets**. It describes how Html elements should be displayed on screen. It is a powerful tool for web designers to change the design and control over web pages that how it should be displayed.

8) Difference between padding and margin?

Answer:

### Padding

Padding is a CSS property that defines the space between an element content and its border (if it has a border). If an element has a border around it, padding will give space from that border to the element content which appears in that border. If an element does not have a border around it, then adding padding has no effect at all on that element, because there is no border to give space from.

So padding is a CSS feature added to elements with borders surrounding them to give space from that border to the element content inside of the border.

### Margin

Margin is a CSS property that defines the space of outside of an element to its next outside element.

Margin affects elements that both have or do not have borders. If an element has a border, margin defines the space from this border to the next outer element. If an element does not have a border, then margin defines the space from the element content to the next outer element.

### Difference Between Padding and Margin

So the difference between margin and padding is that while padding deals with the inner space (the space between an element content and the border), margin deals with the outer space (the space outside of the element) to the next outer element.

Thus, padding and margin have crucial differences that allows them to play key roles in styling a web page.