Assume the client makes a HTTP request based on GET method in the following situations

1. **If our Servlet class contains both doGet(-,-) and doPost(-,-)**
    a. public service(-,-) of HttpServlet
    b. protected service(-,-) of HttpServet
    c. doGet(-,-) of our Servlet class

2. **If our Servlet class contains only doPost(-,-)**
    a. public service(-,-) of HttpServlet
    b. protected service(-,-) of HttpServlet
    c. protected doGet(-,-) of HttpServlet(as our class does not contain doGet(-,-))
    d. 405 response back to client.

3. **If our Servlet class overrides public service(-,-) method, and contains doGet(-,-), doPost(-,-).**
    a. public service(-,-) of our Servlet class [doGet(-,-) of our Servlet class will not get invoked, because the control did not pass on to protected service(-,-) of HttpServlet].

4. **If our Servlet class overrides public service(-,-) method, and does not contain doGet(-,-)**
    a. public service() of our Servlet class
    No 405 response back to client.
5. **If our class overrides public service(-,-) method and it makes a call to super.service(-,-) method, and contains doGet(-,-), doPost(-,-)**
    a. public service(-,-) of our Servlet class
    b. public service(-,-) of HttpServlet(because of super.service(-,-))
    c. protected service(-,-) of HttpServlet
    d. doGet(-,-) of our Servlet class

6. **If our class overrides public service() method, and it makes a call to super.service(-,-) method, and contains only doPost(-,-)**
    a. public service(-,-) of our Servlet class
    b. public service(-,-) of HttpServlet(because of super.service(-,-))
    c. protected service(-,-) of HttpServlet
    d. protected doGet(-,-) of HttpServlet
    e. 405 response back to client

=>In the execution of our Servlet Component don't let the control going to doXxx(-,-) methods of javax.servlet.http.HttpServlet Class because they always generate **405 error response page** indicating our servlet class is totally **incomplete** to process the request.

Q. If both service (-,-) methods are placed in our Servlet Component then which method will be executed?

Ans: Since public service(-,-) method is life cycle method of our Servlet Component only public service(-,-) method will be executed.Even though public service(-,-) method is life cycle method it is recommended to place request processing logic of our Servlet Component by using doXxx(-,-) methods because they are defined based on protocol Http standards ,Moreover if they are not placed properly in our servlet ,the super class doXxx(-,-) methods simply send 405 error to client indicating the problem.

Q. When all the methods of pre-defined HttpServlet Class, are concrete methods why the class itself is given as an abstract class?

Note: In java abstract class can have only abstract methods or only concrete methods or mix of both

- javax.servlet.http.HttpServlet Class is abstract, even though none of the methods with in it are abstract it is because, it contains seven doXxx() methods, to match seven ways of making HTTP request(GET/POST/DELETE/OPTIONS/TRACE/PUT/HEAD). These methods are the request processing methods of a HttpServlet, just like service(-,-) method for a GenericServlet.
- Since there is only one request-processing method in GenericServlet, it is defined as abstract in javax.servlet.GenericServlet Class, which makes the developer making his class to extend javax.servlet.GenericServlet Class, to provide implementation only for one method,But in the case of javax.servlet.http.HttpServlet Class, if all **9 (7+2 methods)** request-processing methods are defined as abstract, then every developer who creates a child class for it, has to provide implementations for all the **9** methods, which is quite an issue. So the specification has made javax.servlet.http.HttpServlet Class to contain implementations for all the 9 methods, but they made the javax.servlet.http.HttpServlet Class itself as abstract, which means no **Developer/ServletContainer** can create an instance of it directly.

<u>The seven http request methods/methodologies</u>

The client can send request to web component of the web application in seven different ways by using seven different Http methods. To process these methods based request in our Servlet Component we can override and use 7 different **doXxx(-,-) methods.**

- **GET (default)**
- **POST**
- **HEAD**
- **PUT**
- **DELETE**
- **TRACE**

- **OPTIONS**

GET:
Default request method, designed to get data from server by generating request without data or with limited amount of data (max. of2kb-8kb).

POST:
Can send request with unlimited amount of data and gathers data from server as response.
The response of GET based or POST based request contains everything including response body like response headers, miscellaneous information, etc…

HEAD:
Same as GET but the HEAD based request related response does not contain response body.
HEAD based requests are useful to test whether web component is present or not.
Note: Even though there are 7 request methods the most regularly used request methods in real world while developing java websites are GET, POST.

PUT:
This is capable of allowing client to place new file or web component in already deployed web application of web server. In real projects after placing websites in the web server of ISP(Internet Service Providers) machine we use FTP(File Transfer Protocol) application from our computers to maintain that website. This FTP application uses PUT method request to add new file or new web resource component in that ISP machine website.

DELETE:
Allows client to send a request having the capability to delete file or web component of web application in the server. FTP application uses this delete method to delete web page or document or anything from the hosted web application of the ISP machine based web server.

TRACE:
This **trace** method request returns all the debugging messages and flow of execution details regarding the request and response of certain web components.

OPTIONS:
The **options** method based request given to web resource component determines using which Http request methods that this servlet can be request from client.
For example:
If our Servlet Component overrides doGet(-,-) method as shown below then the OPTIONS method based request given to the Servlet Component returns the following response
Allow: HEAD, GET, OPTIONS, TRACE
Eg:
public class TestServlet extends HttpServlet

```
{
        public void doGet(HttpServletRequest, HttpServletResponse res)throws
                                        ServletException,IOException
        {
        ……
        ……
        }
 }
```

Note: "POST","PUT" are non-idempotent. "GET","HEAD","OPTIONS","TRACE","DELETE" are idempotent.

Note:

- Since doXxx(-,-) method are given to place 7 types of request processing logic for 7 modes of request given by client and also allows to work with HttpServletRequest, HttpServletResponse object. It is recommended to used doXxx(-,-).
- Using browser we can send only GET or POST methodology request. So while developing regular website we our develop our Servlet Component just having doGet(-,-) and doPost(-,-) method to process the request.
- Hyperlink can generate only GET methodology request where as form page can genate GET or POST methodology . requests, due to this we just use doGet(-,-). doPost(-,-) in our Servlet Component to process the request.
- PUT and DELETE and other request methodologies will be utilized only in the ftp application development which can be used website hosting and web application maintaince.