

## Advance Java programming Assignment 2

1. Which is the connectionless protocol?
  - a) TCP
  - b) UDP
  - c) IP
  - d) HTTP
  
2. What is the port number of HTTP?
  - a) 80
  - b) 23
  - c) 47
  - d) 92
  
3. Which protocol that web browsers and servers use to transfer hypertext pages and images?
  - a) TCP/IP
  - b) UDP
  - c) HTTP
  - d) SMTP
  
4. Originally, all Internet addresses consisted of \_\_\_\_ bit values.
  - a) 32
  - b) 16
  - c) 24
  - d) 8
  
5. What is the full form of DNS?
  - a) Domain Naming System
  - b) Domain Naming Service
  - c) Domain Naming Software
  - d) Domain Naming Security
  
6. What is URL?
  - a) Unicast Resource Location
  - b) Uniform Research Launcher
  - c) Universal Research Locator
  - d) Uniform Resource Locator
  
7. What is the full form of DNS?
  - e) Domain Naming System
  - f) Domain Naming Service
  - g) Domain Naming Software
  - h) Domain Naming Security
  
8. Which class is used to encapsulate both the numerical IP address and the domain name for that address?
  - a) InetAddress
  - b) SocketAddress
  - c) IPAdress
  - d) DNS.Sockets
  
9. Which of the following constructor of InetAddress is generally used?
  - a) InetAddress()
  - b) InetAddress(int port)
  - c) InetAddress(URL url)
  - d) There is no any visible constructor for InetAddress
  
10. Factory method of InetAddress class throws which of the following exception?
  - a) MalformedURLException
  - b) UnknownHostException
  - c) URLException
  - d) MalformedURLExceptionIOException

11. What type of the value that `getAllByName( )` returns?

- a) `InetAddress[ ]`
- b) `InetAddress`
- c) `URL[ ]`
- d) `URL`

12. In which form the `getAddress( )` returns object's internet address?

- a) `int[ ]`
- b) `byte[ ]`
- c) `long[ ]`
- d) `String`

13. Which method returns a string that represents the host name associated with the `InetAddress` object?

- a) `hostName( )`
- b) `myHostName( )`
- c) `giveHostName( )`
- d) `getHostName( )`

14. Which Java networking class is designed to be a listener, and for clients to connect before doing anything?

- a) `Server`
- b) `ServerSocket`
- c) `Servers`
- d) `Socket`

15. What type of the exception might be thrown by a `Socket` class?

- a) `URLException`
- b) `UnknownHostException`
- c) `MalformedURLException`
- d) None

16. What the number 80 specifies in following URL? `http://www.rediff.com:80/index.htm/`

- a) Port number
- b) Site number
- c) URL Uniform number
- d) HTML Page number

17. How to specify the port number to the `URL` class?

- a) Use the method `setPort( )`
- b) Use the method `setValues( )`
- c) Give directly in the constructor
- d) `URL` takes the value automatically

18. Which exception might be thrown by `URL` class?

- a) `IOException`
- b) `UnknownHostException`
- c) `MalformedURLException`
- d) None of the above

19. The `openConnection` is the method of which class?

- a) `URL`
- b) `URLConnection`
- c) `Connection`
- d) `InetAddress`

20. If we create the datagram packet with following constructor:

`DatagramPacket(data, offset, size, address, port)`

What 'size' specifies?

- a) Size of the packet
- b) Maximum data to read
- c) Minimum data to read

**21) Explain the program**

```
import java.net.*;
```

```

class URLEDemo {
    public static void main(String args[]) throws MalformedURLException {
        URL hp = new URL("http://www.osborne/downloads");

        System.out.println("Protocol: " + hp.getProtocol());
        System.out.println("Port: " + hp.getPort());
        System.out.println("Host: " + hp.getHost());
        System.out.println("File: " + hp.getFile());
        System.out.println("Ext:" + hp.toExternalForm());
    }
}

```

## **22) Explain the program**

```

import java.net.*;
class InetAddressTest
{
    public static void main(String args[]) throws UnknownHostException {
        InetAddress Address = InetAddress.getLocalHost();
        System.out.println(Address);
        Address = InetAddress.getByName("google.com");
        System.out.println("add "+Address);
        InetAddress SW[] = InetAddress.getAllByName("google.com");
        for (int i=0; i<SW.length; i++)
            System.out.println(SW[i]);
    }
}

```

## **23) Explain the program**

```

// Demonstrate URLConnection.
import java.net.*;
import java.io.*;
import java.util.Date;

class UCDemo
{
    public static void main(String args[]) throws Exception {
        int c;
        URL hp = new URL("http://www.google.com");
        URLConnection hpCon = hp.openConnection();

        // get date
        long d = hpCon.getDate();
        if(d==0)
            System.out.println("No date information.");
        else
            System.out.println("Date: " + new Date(d));

        // get content type
        System.out.println("Content-Type: " + hpCon.getContentType());

        // get expiration date
        d = hpCon.getExpiration();
        if(d==0)
            System.out.println("No expiration information.");
        else
            System.out.println("Expires: " + new Date(d));

        // get last-modified date
        d = hpCon.getLastModified();
        if(d==0)
            System.out.println("No last-modified information.");
        else
            System.out.println("Last-Modified: " + new Date(d));
    }
}

```

```
// get content length
int len = hpCon.getContentLength();
if(len == -1)
    System.out.println("Content length unavailable.");
else
    System.out.println("Content-Length: " + len);

if(len != 0) {
    System.out.println("=== Content ===");
    InputStream input = hpCon.getInputStream();
    int i = len;
    while (((c = input.read()) != -1) { // && (--i > 0) {
        System.out.print((char) c);
    }
    input.close();

} else {
    System.out.println("No content available.");
}

}
}
```