Q2 (a) Give a detailed note on java creation and evolution.

Answer

Java was conceived by James Gosling , Patrick Nayghton,Chris Warth,Ed Frank and Mike Sheridan at Sun Microsystems in 1991.

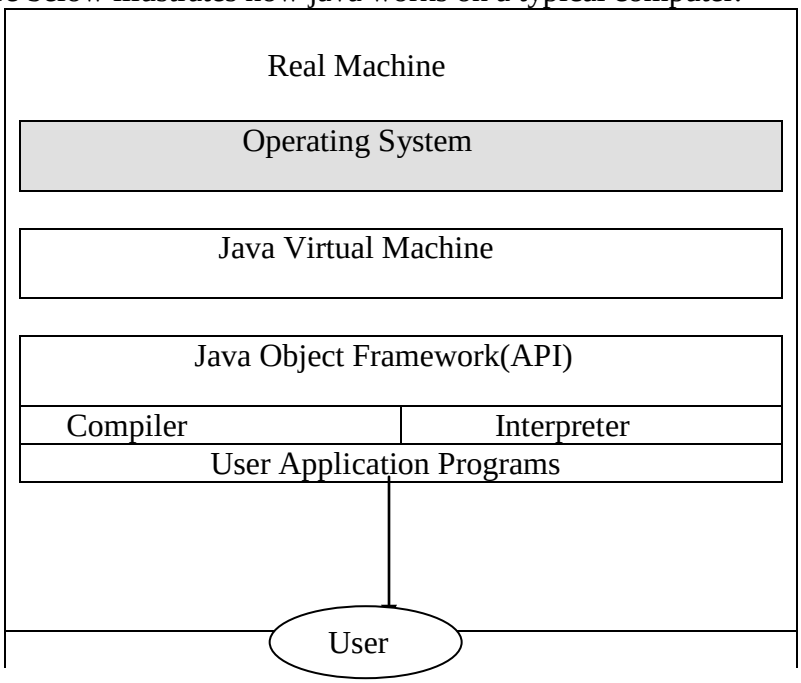
It took 18 months to develop the first working version. This language was initially called "Oak" but was renamed as "java "in 1995.Between the initial implementation of Oak in 1992 and the public announcement of Java in the spring of 1995,many people contributed. With emergence of WWW, java was propelled to the forefront of computer language design, because Web too demanded portable programs. Java is nor upwardly nor downwardly compatible with C++. Java enhanced and refined and the Object-Oriented paradigms used by C++, added integrated support for multithreading, and provided a library that simplified multiple Internet access. Java was to internet programming what C was to system programming: a revolutionary force that changed the world.

Q2 (b) What is JVM? Illustrate with an example.

Answer

All language compilers translate source code into machine code into machine code for a specific computer. Java compiler also does the same thing. Then, how does java achieve architecture neutrality? The answer is that the java compiler produces an intermediate code knows as byte code for a machine that does not exist. This machine is called Java Virtual Machine or JVM.

The figure below illustrates how java works on a typical computer.



The Java Object Framework (java API) acts as the intermediary between the user programs and the virtual machine which in turn acts as intermediary between OS and the Java Object Framework.

Q2 (c) Explain various types of Java selection statements along with suitable examples in Java.

Answer Page Number 77-84 of Text-Book-I

Q3 (a) What are the THREE things performed by Declaration in Java?

Answer

1. It tell the compiler what the variable name is.
2. It specifies what type of data the variable will hold.
3. The place of declaration (in the program) decides the scope of variable.

Q3 (b) What is an interface? Explain how to implement an interface in Java.

Answer

An Interface is nothing but a contract as to how a class should behave. It just declares the behavior as empty methods and the implementing class actually writes the code that will determine the behavior.

When you implement an interface, you're agreeing to adhere to the contract defined in the interface .That means you're agreeing to provide legal implementations for every method defined in the interface, and that anyone who knows what the interface methods look like can rest assured that can invoke those methods on an instance of your implementing class. (They need not bother much about how you have implemented it. All they bother about is whether a method of the name mentioned in the interface is available or not)

You cannot successfully implement an interface without providing method implementation for all the methods declared inside the interface. This how the java system ensures that when someone know a certain method name in an interface and has an instance of a class that implements it, can actually call that method without fear that the method isn't implemented inside the class.

Assuming an interface, Convertible, with two methods:OpenHood(),and
set OpenHoodFactor(),the following class will compile:

```
public class Ball implements Convertible {//Keyword 'implement'
public void openHood(){}
public void set OpenHoodFactor(int bf){}
}
```

Though it compile and runs as well, it is actually doing nothing …..the interface contract guarantees that the class implementing it will have a method of a particular name but it never guaranteed a good implementation . In other words, the compiler does not bother whether you have code inside your

method or not .All it cares is if you have methods of the matching names as in the interface. That's all……..

Implantation classes must adhere to the same rules for method implementation as a class extending an abstract class. In order to be a legal implementation class, a non abstract implementation class must do the following:

- Provide concrete (non abstract) implementations for all methods from the declared interface.
- Follow all the rules for legal overrides.
- Declare no checked exceptions on implementation methods other then those declared by the interface method, or subclasses of those declared by the interface method.
- Maintain the signature of the interface method, and maintain the same return type (or a subtype)
- It does not have to declare the exceptions declared in the interface method declaration.

Q3 (c) Explain the concept of Inheritance in Java.

Answer

Java classes can be reused in several ways. This is done creating new by new classes reusing the properties of existing onse.The mechanism of deriving a new class from an old one is called inheritance. The old class is called the base class or parent class or super class and the new one is called the subclass or derived class or child class.

The inheritance allows subclasses to inherit all the variables all the methods of their parent classes. Inheritance may take several different forms:

1. Single inheritance
2. Multiple inheritance
3. Hierarchical inheritance
4. Multilevel inheritance

Q4 (a) Explain the mistake in the following program. What changes would you recommend?

```
class ThrowsDemo
        {
            static void throwOne( )
             {
                    System.out.println("Inside throwOne.");
                    Throw new IllegalAccessException("demo");
             }
             public static void main(String args[])
             {
                     throwOne ( );
             }
        }
```

Answer

This program tries to throw an exception that it does not catch . Because the program does not specify a throws clause to declare this fact, The program will not compile. To make the above program correctly ,first ,we need to declare that throwOne() throws illegalAccessException. Second, main () must define a try/catch statement that catches this exception.

The corrected Code is:

```
class ThrowsDemo
        {
            static void throwOne( )throws illegalAccessException
             {
                    System.out.println("Inside throwOne.");
                    Throw new IllegalAccessException("demo");
              }
              public static void main(String args[])
            {   try
              {
                     throwOne ( );
              }

          catch(illegalAccessException e)
       {
                 system.out.println("caught"+e);


        }
}
}
```

Q4 (b) Is it necessary that each try block must be followed by a catch block? Explain.

Answer

It is not necessary that each try block must be followed by a catch block .It should be followed by either **a** catch block OR a finally block. And whatever exceptions are likely to be thrown should be thrown should be declared in the throws clause of the method.

Q4 (c) What is the basic difference between the given two approaches to exception handling: "try catch block" and "specifying the candidate exceptions in the throws clause"? When should you use which approach?

Answer

In the first approach as a programmer of the method, you yourself are dealing with the exception. This is fine if you are in a best position to

decide should be done in case of an exception. Whereas if it is not the responsibility of the method to deal with it's own exceptions, then do not use this approach. In this case use the second approach.

Q5 (a) Write a code in Java for a tiny text editor. The steps involved would be creation of an array of String objects, reading lines of text, storing each line in the array. It should read up to 100 lines or until you enter "stop".

Answer

```
//A tiny editor
import java.io.*;
class TinyEdit
{
public static void main(String args[])
throws IOException
{
//creat a BufferedReader using System.in
BufferedReader br=new
BufferedReader(new InputStreamReader(System.in));
String str[]=new String[100];
System.out.println("Enter lines of text.");
System.out.println("Enter'stop'to quit.");
For(int I=0;I<100;I++)
{
str[i]=br.readLine();
if(str[i].equals("stop"))break;
}
System.out.println("\nHere is your file:");
//display the lines
For(int I=0;I<100;I++)
{
if(str[i],equals("stop"))break;
System.out.println(str[i]);
}
}
}
```

Q6 (a) List any four interfaces along with brief description that underpin collections in Java.

Answer

Interface	Description
Collection	Enables you to work with group of Objects; it is at the top of the collections

	herachy.
Desque	Extends Queue to handle a double-ended queue. (Added by java SE 6)
List	Extends Collection to handle sequences (lists ob objects)
Navigational Set	Extends SortedSet to handle retrieval of elements based on closestmatch searches (Added by java SE6).
Queue	Extends collection to handle special types of lists in which elements are removed only from the head.
Set	Extends collection to handle sets , which must contain unique elements.
SortedSet	Extends Set to handle sorted sets

Q6 (b) Discuss swing packages for Java programs

Answer Page Number 863,879,889 of Text-Book-I

Q7 (a) What are the three major components of DNS and how are they related to the three layers or views of the domain system?

Answer

The DNS has three major components:

The DOMAIN NAME SPACE AND RESOURCE RECORDS, which are specifications for a tree structured name space tree names a set of information, and query operations are attempts to extract specific type of resource information that is desired. For example, the internet uses some of its domain names to identify hosts; queries for address resources return Internet host addresses.

NAME SERVERS are server programs which hold information about the domain tree's structure and set information. A name server may cache structure or set information about any part of the domain tree, but in general a particular name server has complete information about a subset of the domain space, and pointers to other name servers know the parts of the domain tree for which they have complete information; a name server is

said to be an AUTHORITY for these parts of the name space. Authoritative information is organized into units called. Zones and these zones can be automatically distributed to the name servers which provide redundant service for the data in a zone.

RESOLVERS are programs that extract information from name server in response to client requests. Resolvers must be able to access at least one name server and use that name server's information to answer a query directly, or pursue the query using referrals to other name servers. A resolver will typically be a system routine that is directly accessible to user programs; hence no protocol is necessary between the resolver and the user program .

These three components roughly correspond to the three layers or views of the domain system:

From the user's point of view, the domain system is accessed through a simple procedure or OS call to a local resolver. The domain space consists of a single tree and the user can reqest information from any section of the tree.

From the resolver's point of view, the domain system is composed of an unknown number of name servers. Each name server has one or more pieces of the whole domain tree's data, but the resolver view each of these databases as essential static.

From a name server's point of view, the domain system consists of separate sets of locals' information called zones. The name server has local copies of some zones. The name server must periodically refresh its zones from master copies in local files or foreign name server. The name server must concurrently process queries that arrive from resolves

In these interests of performance, implementation may couple these functions. For example a resolver on the same machine as a name server might share a database consisting of the zones managed by the name server and the cache managed by the resolver

Q7 (b) Explain the core attributes of XHTML with examples..

Answer

Class

This attributes indicates the class or classes that a particular element belongs to. A class name might be used by a style sheet to associate style rules to multiple elements at once. For example, one could associate a special class name called"impotant" with all elements that should be rendered with a yellow background. Class values are not unique to a particular element, so<b class=" important"> could be used as well as<p class="important"> in the same document.

Id

This attribute specifies a unique alphanumeric identifier to be associated with an element. Naming an element is important to being able to access it with a style sheet, a link, or a scripting language. Name should be unique to a

document and should be meaningful, so although id =”x1” is perfectly valid id =”paragraphe1” might be better. Values for the id attributes must begin with a letter (A-Z and a-z) and may be followed by any number of letters, digits, hyphens and periods. Of course it is generally not encouraged to have id values with special characters, even if they are allowed.

One potential problem with the id attribute is that for some elements, particularly form controls and images, the name attributes already serves its function. Values for name should not collide with values for id, as they share the same naming space.

For example, the following would not be allowed;

this is a test.

Style

This attributes specifies an inline style associated with an element, which determines the rendering of the affected elements. Because the style attributes allows style rules to be used directly with the element, it gives up much of the benefit of style sheets that divide the presentation of an HTML document from its structure .An example of this attribute’s use is shown here:

An example of this attribute’s use is shown here:

<strong style=”font-family:Arial;
font-size:18pt”>Important text</>

title

The title attributes supplies advisory text that can be rendered as a Tooltip when the mouse is over the element.(Internet Explorer supports this Tooltip display, but Netscape browsers perior to version 6 do not) In some case , like the a element , the title attributes can provide additional help in bookmarking. Like the title for the document itself , title attributes values such as advisory information should be short , yet useful.

For example

<p title =”paragraph1”>provides little information of value, wheres<p tittle =”HTML: the complete Reference Appendix A paragraph 10”>provides nuch detail. when combined with scripting , this can provide facilities for automatic index generation.

Q8 (a) Describe the concept of Server Side Include. Elaborate how this concept is used with files located in the same directory and files located in a different directory?

Answer

A server – side is a coding that you can include within your HTML document that will tell the web server to include other information with the document being served.

Server side includes are a handy way to include information on your pages automatically.
For example , you can use an #include statement to display the header and footer of your pages automatically and make sure that each page display the same information you can save time by using a single header.html or footer.html files if this information changes . You can also use server side includes for graphics such as logos or images maps that should appear on multiple pages.

You will need to use the .html extension for pages using the server side include(those that have # include statements) for accounts hosted on web server .The files that is being include can have other extensions , such as .html.

Please note that for security reasons, the EXEC server side include is NOT enabled on Web server. The EXEC command is often used to run a cgi or perl program from a web page , but this function is not enabled on web server.

Using Server side include with files in the same directory
To use a server side include, use the # include statement . This command tells the server to include another files,and tells it the files name that should be included:
<!--#include FILE="filename.html"--->
this page uses server side includes to display the webmaster banner and header information . An example of the command used is:

<!--# include file="header.html"---->
In thbis example it is placed in the body of the document, immediately after the <BODY> tag. One thing about SSI includes is the process of including. It take two separate files and create one whole files. What the server does is actually to take the ssi page and physically inserted it into the page calling it.
Using server side includes with files located in different directories

If the server side includes calls a file that is located in another directory in the account, you need to make some modifications. For example :<!--#include virtual ="../header.html"--->
In the above example, the file named "header.html" is located in the directory above the location of the files referencing it. Also instead of using "#include file" you must use the command "#include virtual"
Footer
The footer for this page is created dynamically, so that the date that the file was last modified automatically appears in the footer. The server side includes statement calles another .shtml file, using this code:
<!--#include file="footer.shtml"--->
The footer statement on this page is placed at the bottom of the files, just before the </BODY>tag.

Q8 (b) A common mistake is to begin creating Web pages before doing sufficient groundwork on the information blueprint for a site. What you need to is to create the site's Information Architecture or IA.

List six concrete steps to IA. Specify the questions which may help you reveal the true objectives of the site.

Answer

1. The six concrete steps to IA are :
2. Define goals
3. Define audience
4. Create and organize content
5. Formulates visual presentation concepts
6. Develop sitemap and navigation , and
7. Design and produce visual forms

Here are sample question which may help you reveal the true objectives of the site:

1. What is the mission or purpose of the organization? Read the mission statements and business plans. Review client‘s literature. Remember, client’s mission may change with time.
2. What are the short and long – term goals of the site? Key people may not be thinking in the long term. Immediate need may be to get the site up and running. Look toward the future; accommodate growth and change.
3. Who are the intended audiences? Inadequate analysis in this area is the number one mistake made in designing sites!
4. Why will people come to your site? For the first time? For repeat visits?
5. Does the site provide a well – defined services? Or sell specific products?
6. Does your client have an existing site? Find this out now.

Q9 (a) Describe the outline and structure of a CGI Program.

Answer

A CGI program(or script) is invoked by the web server which provides input to the CGI program and receive its output . Typically, a CGI program follows this outline:

1. Determines request method and receives input data: The request method is given by the REQUST_METHOD environment variable. For a POST query the data is read from standard input and the length of the data is indicated by the CONTENT LENGTH environment variable .for a GET query, the input data is the value of the QUERY STRING environment variable.
2. Decodes and checks input data: the form –url encoded input data is decoded and the key- value pairs recoverse. The correctness of the input data is checked. Incomplete or incorrect input results in a response to the end-user for the correct information.
3. Performs tasks: The input data is complete and correct. The program now processes the information and perform required actions.
4. Produces output: A generated response is sent to standard output. The response is usually in HTML format.

Q9 (b) Elaborate four major categories of Event attributes for elements that are used to connect events to handlers.

Answer

Event attributes for elements that are used to connect events to handlers

1. Window Events-Onload (page loaded) and on unload (page unloaded)attributes are available in body and frame set elements.
2. Mouse Events-Onclick(left mouse button clicked), ondblclick(left mouse button double clicked), onmousedown(left mouse button pressed), onmousemove(left mouse button released) ,onmousemove (mouse moved), onmousedover (mouse over element), and onmouseout (mouse moved), ommouseover(mouse over element), and onmouseout(mouse left element) attributes are common to most HTML4.0elements . But some versions of javascript may only support these in form elements and links.
3. Form Element Events-onfocus (the element gained focus), onblur (the element lost focus), onchange (content of element changed),onsubmit(form submitted), onreset (form reset), and onselect(element content is selected) attributes are for all form related elements .
4. Keyboard Events –onkeydown (a keyboard key is being pressed),onkeypress(a key is pressed), and onkeyup(a key is released) are common to most HTML4.0

Text Books

1. The Complete Reference Java, Herbert Schildt, TMH, Seventh Edition, 2007.

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