Q.2 a. How does Java differ from C and C++?

Answer: Page 16 of Text Book 1

b. Explain the meaning of the following statement public static void main (string args [ ])

Answer: Page 26 of Text Book 1

Q.3 a. What are the three things performed by declaration?

Answer:

1. It tells 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**.**

b. List any 7 Java operators with description, associatively and relative ranking.

Answer:

S. No.	Operator	Description	Associatively	Relative Rank
1	==	Equality	Left to right	1
2	!=	Inequality	Left to right	1
3	&	Bitwise AND	Left to right	2
4	^	Bitwise XOR	Left to right	3
5	|	Bitwise OR	Left to right	4
6	&&	Logical AND	Left to right	5
7	||	Logical OR	Left to right	6
8	? :	Conditional operator	Right to left	7
9	=	Assignment Operators	Right to left	8
10	op=	Shorthand Assignment	Right to left	9

Q.4 a. Explain the concept of Method Overloading in Java with explain.

Answer:

Method overloading allows java programmer to declare method with same name but different behavior. Method overloading is based on polymorphism in java. In case of method overloading, method with same name co-exists in same class but they must have different method signature, while in case of method overriding is resolved using static binding in java at compile time while method overriding is resolved using dynamic binding in java at runtime. in short when you overload a method in java its method signature got changed while in case of overriding method signature remains same but a method can only be overridden in sub class. Since java supports polymorphism and resolve object at run-time it is capable to call overridden method in java.

b. What are vectors? How are they advantageous over arrays?

Answer: Page 166 of Text Book

c. Explain 'equals ()' function of the string class.

Answer: Page 163 of Text Book

Q.5a. How to extend an interface? Illustrate with an example.

Answer:
An interface can extend another interface, similarly to the way that a class can extend another class. The extends keyword is used to extend an interface, and the child interface inherits the methods of the parent interface.

The following Sports interface is extended by hockey and Football interfaces.

```
//Filename: Sports.java
Public interface Sports
{
  Public void setHome Team (String name);
  Public void setVisiting Team (String name);
}

//Filename: Football.java
Public interface Football extends Sports
 {
    Public void home Team Scored (int points);
    Public void visiting Team Scored (int points);
    Public void end Of Quarter (int quarter);
}
//Filename: Hockey.java
 Public interface Hockey extends Sports
{
  Public void home Goal Scored();
  Public void visiting Goal Scored();
  Public void end Of Period(int period);
  Public void overtime Period (int ot);
}
```

b. List any three frequently used packages in Java with descriptions.

Answer:

- java.lang — basic language functionality and fundamental types
- java.util — collection data structure classes
- java.io — file operations
- java.math — multiprecision arithmetics
- java.nio — the New I/O framework for Java
- java.net — networking operations, sockets, DNS lookups, ...
- java.security — key generation, encryption and decryption
- java.sql — Java Database Connectivity (JDBC)
- java.awt — basic hierarchy of packages for native GUI components
- javax.swing — hierarchy of packages for platform-independent rich GUI components
- java.applet — classes for creating an applet

Q.6 a. Here's a piece of code that has been saved in a file called "JollyMessage.java":

```
     // A jolly message is written to the screen!
      class Jollymessage
     {
      public static void main(String[] args) {
 //Write the message to the terminal window
 System.out.println("Ho Ho Ho!");
}
      }
```

The above code will produce a runtime error message. To put it another way, a mistake has been made but it won't be picked up when the program is compiled, only when it is run.

What is the mistake and what error will be returned when the compiled code is executed? How can the problem be rectified?

Answer:

- The class is called "Jollymessage" whereas the filename is called "JollyMessage.java"

- The compiler won't complain because technically there is nothing wrong with the code. It will create a class file that matches the class name exactly (i.e., Jollymessage.class). When you come to run the program called JollyMessage, you can't because there is no file called JollyMessage.class.

- The error we get when you try and run a program with the wrong name is:

- Exception in thread "main" java.lang.NoClassDefFoundError: JollyMessage (wrong name: JollyMessage).

- As we know Java is case sensitive. We just need to change the class name so that it matches the file name (or vice versa).

b. Discuss the concept of streams in java. What are the two basic streams?

Answer:

A stream is data that you access in sequence. You could think of it like a train that you watch from a tunnel entrance so that you can just see one car at a time. Or a stream of widgets coming across a conveyor belt requiring you to tighten a screw on each one before it passes by to the next person down the assembly line who has to pound it with a hammer, and so on. Or sticks floating down a river while you watch from a bridge.

InputStream and OutputStream are the basic stream classes in Java, one letting us bring data in from somewhere, the other letting us put data out to somewhere. All the other streams just add capabilities to the basics, like the ability to read a whole chunk of data at once for performance reasons (BufferedInputStream) or convert from one kind of character set to Java's native unicode (Reader), or say where the data is coming from (FileInputStream, ServletInputStream, SocketInputStream, ByteArrayInputStream, etc.)

The various classes are meant to wrap each other, so we can build, for example, a BufferedInputStream that reads from a file:

```
InputStream in = new BufferedInputStream(new FileInputStream(myfile));
```

Then if you decide you want to read data from a network socket instead, you can just change FileInputStream() to SocketInputStream() (and add a bit of code to establish the network connection of course) and you won't have to change anything else in your program.

Q7 a. What are name servers? Explain.

Answer:

Nameservers, special computers accessible on the Internet, tell where a website is located. Nameservers are responsible for resolving, a special process in which a domain name matching your website is taken and a physical address is obtained.

The resolving process begins when a visitor goes to your website. The visitor enters your domain name in the Internet browser. The browser then searches the nameservers about the location of the server to find out just where your website is located. Then, it connects to the server and opens your web page. The whole process is similar to looking through a phone book. For example, when you pick up the phone book to look for the name John Doe, it is like the beginning of the resolving process – just like typing a name into the Internet browser. When you find the phone number, you have resolved the process, just like a nameserver helps find a website address!

While the resolving process is not something you as the website developer must complete or even need to understand, you do need to know some of the information. You will need to provide the names of the nameservers when you buy or set up your domain name. Your web host will assign your nameservers when you enroll in their hosting service. Other than that, let the computer and name server handle the rest. All you have to do is get your web page designed and on the World Wide Web!

b. Are there any problems spacing images in toolbars? What is the solution?

Answer:

Some browsers insert a space between adjacent elements if there is any whitespace between the end-tag of one element and the start-tag of the other, which can be annoying if you're trying to put together a tool bar or a compound image.

Thus

```
<p><img src="button1.gif" alt='Button1">
<img src="button1.gif" alt='Button1"></p>
```

may be rendered as

[Button1] [Button2]

rather than

[Button1][Button2]

If you really want to ensure that two images are contiguous, the only way is to accomplish it is do something like

```
<p><img src="button1.gif" alt='Button1"><img
src="button1.gif" alt='Button1"></p>
```

Q8. a. How can you control the horizontal and vertical positioning of content within a cell? Explain through examples.

Answer:

We can control the horizontal and vertical positioning of content with a cell by using the align and valign attributes, respectively. The scope of entries affected by the attribute depends on which table element has the attribute.

Horizontal alignment

You can set horizontal alignment of cell content by setting the align attribute to any of these values:

left

center

right

char

justify

A value of char means the content should line up according to a certain character in the cell data, typically a decimal character. Then a column of numbers with decimals will have the decimals lined up. If you set align="char" for a column, you need to specify a char="." attribute (or other character) as well to tell the processor on what character to line up the entries.

The align attribute can be set on any of the following elements, so it applies to the scope of that element:

tgroup

colspec

spanspec

entry

For example, you could set align="left" in tgroup for the whole table (if you have one tgroup). Then you can set align="center" in one colspec to center all entries in that column. The following is an example.

```
<table><title>My table</title>

<tgroup cols="4" align="left">

<colspec colnum="1" colname="col1" colwidth="1*" />

<colspec colnum="2" colname="col2" colwidth="2*" align="center" />

<colspec colnum="3" colname="col3" colwidth="1.5*"/>
```

```
<colspec colnum="4" colname="col4" colwidth="1*" align="char" char="."/>

<thead>

...
```

In this example, the tgroup element sets the default alignment for the table to left. Entries in column 2 will instead be centered, and entries in column 4 are to be aligned on the decimal character in the cell data.

Vertical alignment

You can set vertical alignment of cell content by setting the valign attribute to any of these values:

top

middle

bottom

The valign attribute can be set on any of the following elements so it applies to the scope of that element:

thead

tbody

tfoot

row

entry

For example, you could set valign="top" in tbody for the whole table. Then you can set valign="middle" in one row to middle-align all cells in that row, or on a single entry to middle-align only that cell. The following is an example.

```
<table><title>My table</title>

<tgroup cols="4" >

...

<thead valign="bottom">
```

```
...

<tbody valign="top">

  <row valign="middle">

    <entry>...</entry>

    <entry valign="top">...</entry>

...
```

In this example, the row alignment overrides the tbody alignment, and then the entry alignment overrides the row alignment.

b. What are the elements of design? Explain any four.

Answer:

LINE
Line can be considered in two ways. The linear marks made with a pen or brush or the edge created when two shapes meet.

SHAPE
A shape is a self contained defined area of geometric or organic form. A positive shape in a painting automatically creates a negative shape.

DIRECTION
All lines have direction - Horizontal, Vertical or Oblique. Horizontal suggests calmness, stability and tranquillity. Vertical gives a feeling of balance, formality and alertness. Oblique suggests movement and action

SIZE
Size is simply the relationship of the area occupied by one shape to that of another.

TEXTURE
Texture is the surface quality of a shape - rough, smooth, soft hard glossy etc. Texture can be physical (tactile) or visual.

COLOUR
Also called Hue

VALUE
Value is the lightness or darkness of a colour. Value is also called Tone

Q9 a. Explain the essential parts of a typical form.

Answer: Page 279 of text Book 1

b. Describe how to embed JavaScript in a Web Page.

Answer:

A Javascript program usually consists of various code parts that cooperate to do the job. Where and how to place code in a Web page depend on the purpose of the code. But generally, the code can appear either in <script> elements or as values of event-handling attributes of HTML tags. Any number of script elements can be placed in head, body, block and inline elements. Place a script in the head element unless it generates document content, as in Ex: Date.

1. Code for defining functions or creating data structures are placed in <script> elements inside the page's <head> element.
2. Code for generating HTML to be displayed as part of the page are placed in <script> elements inside the <body> where the generated code will be inserted.
3. Code for actions triggered by events are given as values of event attributes of HTML tags. The code is usually a function call or a few short statements. The onfocus,onblur, onclick, onmouseover, and onmouseout are examples of event attributes.

A Javascript program often works for not one but a set of pages. In that case, it is better to place the code in a separate file. The file can then be attached to a Web page as follows:

```
<script type="text/javascript" src="file.js"></script>
```

The src attribute of <script> gives the URL of the program file. Note we used the conventional .js file name suffix. With file inclusion, any corrections or improvements in the .js file will be reflected in multiple pages automatically. Furthermore, a browser will download the program only once, reducing download time for these pages. Older browsers that do not support client-side scripting will ignore the <script> tag and the program file will not be included. If the code is placed in the <script> tag, then the following construct is often used to avoid the code being mistaken as page content by older browsers:

```
<script type="text/javascript">
<!--
var d = new Date;
...
// -->
</script>
```

<u>Text Books</u>

(1) Programming with Java - A primer, E. Balagurusamy , 3rd Edition, TMH, 2007

(2) An Introduction to web Design + Programming, Paul S.Wang & Sanda S. Katila, Thomson course technology, India Edition, 2008