Q.2 a. What do you understand by platform-independence and how is it implemented by Java?

Answer: Page Number 13 of Text Book I

b. How Java is associated with the internet?

Answer: Page Number 17 of Text Book I

Q.3 a. Explain Numeric and Non-Numeric data types of Java.

Answer: Page Number 46 of Text Book I

b. Explain how Java takes care of type conversion and casting with examples.

Answer:

Java's Automatic Conversions:

When one type of data is assigned to another type of variable, an *automatic type conversion* will take place if the following two conditions are met:

- The two types are compatible.
- The destination type is larger than the source type.

When these two conditions are met, a *widening conversion* takes place. For example, the **int** type is always large enough to hold all valid **byte** values, so no explicit cast statement is required.

For widening conversions, the numeric types, including integer and floating-point types, are compatible with each other. However, the numeric types are not compatible with **char** or **boolean**. Also, **char** and **boolean** are not compatible with each other.

Casting Incompatible Types:

To create a conversion between two incompatible types, you must use a cast. A *cast* is simply an explicit type conversion. It has this general form:

(*target-type*) *value*

Here, *target-type* specifies the desired type to convert the specified value to. For example, the following fragment casts an **int** to a **byte**. If the integer's value is larger than the range of a **byte**, it will be reduced modulo (the remainder of an integer division by the) **byte**'s range.

```
int a;
byte b;
// ...
b = (byte) a;
```

Q.4 a. Explain Method overloading and overriding with example for each.

Answer: Page Number 134 &142 of Text Book I

b. How are two dimensional arrays declared in Java? Write a program to demonstrate the same.

Answer:

To declare a multidimensional array variable, specify each additional index using another set of square brackets. For example, the following declares a two-dimensional array variable called **twoD**.

```
        int twoD[][] = new int[4][5];

// Demonstrate a two-dimensional array.
class TwoDArray {
        public static void main(String args[]) {
                int twoD[][]= new int[4][5];
                int i, j, k = 0;
                        for(i=0; i<4; i++)
                                for(j=0; j<5; j++) {
                                        twoD[i][j] = k;
                                        k++;
                                }
                        for(i=0; i<4; i++) {
                                for(j=0; j<5; j++)
                                        System.out.print(twoD[i][j] + " ");
                        System.out.println();
                        }
        }
}
```

Q.5 a. How do we define interfaces in Java? Explain with the help of an example.

Answer: Page Number 182, 184 of Text Book I

b. Explain the one method for creating threads in Java.

Answer:

In the most general sense, we can create a thread by instantiating an object of type Thread.

Java defines two ways in which this can be accomplished:

- You can implement the **Runnable** interface.
- You can extend the **Thread** class, itself.

c. What are the benefits of organizing classes into packages?

Answer: Page Number 192 of Text Book I

Q.6 a. Explain throw, throws and finally clauses of java exceptions.

Answer:

It is possible for your program to throw an exception explicitly, using the **throw** statement. The general form of **throw** is shown here:

throw *ThrowableInstance*;

Here, *ThrowableInstance* must be an object of type **Throwable** or a subclass of **Throwable**. Simple types, such as **int** or **char**, as well as non-**Throwable** classes, such as **String** and **Object**, cannot be used as exceptions. There are two ways you can obtain a **Throwable** object: using a parameter into a **catch** clause, or creating one with the **new** operator.

If a method is capable of causing an exception that it does not handle, it must specify this behavior so that callers of the method can guard themselves against that exception. You do this by including a **throws** clause in the method's declaration. A **throws** clause lists the types of exceptions that a method might throw. This is necessary for all exceptions, except those of type **Error** or **Runtime Exception**, or any of their subclasses. All other exceptions that a method can throw must be declared in the **throws** clause. If they are not, a compile-time error will result.
This is the general form of a method declaration that includes a **throws** clause:

type method-name(parameter-list) throws *exception-list*
{
// body of method
}
Here, *exception-list* is a comma-separated list of the exceptions that a method can throw.

finally creates a block of code that will be executed after a **try**/**catch** block has completed and before the code following the **try/catch** block. The **finally** block will execute whether or not an exception is thrown. If an exception is thrown, the **finally** block will execute even if no **catch** statement matches the exception. Any time a method is about to return to the caller from inside a **try/catch** block, via an uncaught exception or an explicit return statement, the **finally** clause is also executed just before the method returns. This can be useful for closing file handles and freeing up any other resources that might have been allocated at the beginning of a method with the intent of disposing of them before returning. The **finally** clause is optional. However, each **try** statement requires at least one **catch** or a **finally** clause.

b. Write a program in Java to copy one file into another.

Answer: Page Number 300-304 of Text Book

Q.7 a. Which are the different content types available for web? Explain each.

Answer:

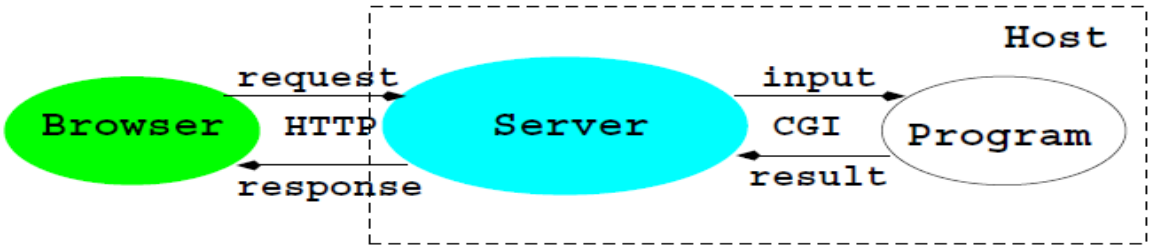
On the Web, many different types of files can be placed and retrieved. The Web server andWeb browser use a set of standard designations to indicate file content types in order to process diffrent files correctly.
The Web borrowed the content type designations from the Intenet email system and use the same MIME (Multipurpose Internet Mail Extensions) defined content types. There are hundreds of content types in use today.
Many popular types are associated with standard file extensions.
When a Web server returns a document to a browser, the content type is indicated. The content type information allows browsers to decide how to process the incoming content. Normally, HTML, text, GIF, JPEG, PNG etc. are handled by the browser directly. Others types such as quicktime, PDF, audio and video are handled by plug-ins or helper programs.

b. How are web pages generated dynamically? Explain.

Answer:

Documents available on the Web are usually prepared and set in advance to supply some fixed content, either in HTML or in some other format such as plain text, GIF, or JPEG. These fixed documents are static. A Web server can also generate documents on-the-fly that bring these and other advantages:

- customizing a document depending on when, where, who, and what program is retrieving it.
- collecting user input (with HTML forms) and providing responses to the incoming Information
- enforcing certain policies for outgoing documents
- supplying contents such as game scores and stock quotes, that are changing by nature.



Q.8 a. Explain how special symbols ©, ®, etc. can be inserted in a web page using HTML.

Answer: Page Number 71-72 of Text Book

Q.9 a. Explain the HTTP framework briefly.

Answer:

On the Web, browser-server communication follows the HTTP protocol. It is good for a Web developer to have a basic understanding of HTTP. Here is the framework of an HTTP transaction:

1. Connection|A browser (client) opens a connection to a server.
2. Query|The client requests a resource controlled by the server.
3. Processing|The server receives and processes the request.
4. Response|The server sends the requested resource back to the client.
5. Termination|The transaction is done and the connection is closed unless another transaction will take place immediately between the client and server.

b. What is the outline of a CGI program? Briefly explain.

Answer: Page Number 295 of Text Book

Text Books

1. Programming with Java- A primer, E. Balagurusamy, Third Edition, TMH, 2007.
2. An Introduction to Wed Design + Programming, Paul S. Wang and Sanda S. Katila, Thomson Course Technology, India Edition, 2008.