9.9 Creation of files

For creating a file following information is required.

1. **Suitable name for the file**

   It is unique string of characters used to identify a file on the disk.
   A file name contains two parts one is file name separated by period with extension.
   Input.c, input java

2. **Data types to be stored**

   It is important to decide the type of file stream classes to be used for handling the data.

3. **Purpose of file**

   Purpose of file is reading, writing or updating.

4. **Methods of creating file**

   - ➢ It must be opened first. This is done by creating a file stream and then linking it to the filename.
   - ➢ A file stream can be defined using the classes of Reader/Inputstream for reading data and Write/OutputStream for writing data.
   - ➢ The common stream classes used for I/O operations.
   - ➢ The constructors of stream classes used to assign the desired filename to the file stream objects.

Source or **Destination**	Character		Bytes	
	Read	Write	Read	Write
Memory	CharArrayReader	CharArray Writer	ByteArrayInputStream	ByteArrayOutputStream
File	FileReader	FileWrite	FileInputStream	FileOutputStream

Pipe	PipedReader	PipedWriter	PipedInputStream	PipedOutputStream

Fig. Common Stream Classes Used for I/O Operations

- There are two ways for initialization of the file stream objects:

1) Direct Approach:

The file name provide directly. The following code segment shows the use of direct approach.

```
FileInputStream fis; //Declare file stream object try
{
// Assign the filename to file stream
Object fis = new FileInputStream ("ABC.txt");
.......
}
Catch (IOException ())
   .......
   .......
```

2) Indirect approach

This approach uses a file object that has been initialized with the desired filename, it illustrated by the following code:

```
...
...
File inFile; //Declare a file object
InFile = new File ("ABC.txt");//Assign the filename to file object
FileInputStream fis;
try{        // give value of the file object to // the file stream object
fis = new FileInputStream (inFile);
...
}
Catch (....)
...
...
```