

USING LIBRARY CLASSES

Simple input, output. String, static variables and static methods, packages and import statements.

Q. What is the difference between byte oriented IO and character oriented IO? How are these two performed in Java?
OR

Q. What are the two ways of obtaining input in Java?

Ans: Byte oriented IO reads bytes of data or binary where there is no notation of datatypes. Character oriented IO on the other hand performs IO which is specially character oriented. In Java byte oriented IO is performed through data streams where

as character oriented IO is performed through Readers and Writers.

Q. What is an Exception?

Ans: Exception in general refers to some contradictory or unusual situation which can be encountered while executing a program.

Q. What is exception and what is exception handling?

Ans: During program development there may be some cases where the programmer does not have the certainty that this code-fragment is going to work right, either because it accesses resources that do not exist or it goes out of range. These types of anomalous situations are generally called exception and the way to handle them is called exception handling.

Q. What are the advantages of Exception Handling?

Ans: (i) Exception handling separates error handling code from normal code. (ii) It

clarifies the code and enhanced readability. (iii) It stimulates consequences as the error handling takes place at one place and in one manner. (iv) It makes for clear, robust, fault tolerant programs.

Q. When is Exception Handling required?

Ans: The exception handling is ideal for:

- (i) Processing exceptional situations.
- (ii) Processing exceptions for components that handle them directly.
- (iii) Processing exceptions for widely used components that should not process their own exception.

Q. What do you mean by try block? How do you define it, give an example.

Ans: The try block is the one that contains the code that is to be monitored for the occurrence of an exception. A try block is defined by enclosing the statements that might possibly raise an exception in. For example if the formatting exception are to be handled while an integer is being read

from the keyboard, then the following try block can be used:

```
int inData;  
BufferedReader br=new BufferedReader(  
new InputStreamReader(System.in));  
try  
{  
inData=Integer.parseInt(br.readLine());  
}
```

Q. What do you mean by catch block?
How do you define it, give an example.

Ans: The catch block is the one that contains the code handle an exception. It must follow the try block. i.e. there should be no statement between the try and the catch blocks. If the catch block is written for the above try block then we may do it as follows:

```
int inData;  
BufferedReader br=new BufferedReader(  
new InputStreamReader(System.in));  
try  
{
```

```
inData=Integer.parseInt(br.readLine());
}
catch(NumberFormatException nfEx)
{
System.out.println("Input format is
incorrect");
}
```

Q. What is finally block? When and how it is used.

Ans: The finally block is one of the exception handling blocks. The code written in this block is always executed irrespective of whether an exception was reported or not, or even if it was handled successfully or not. The purpose of this block is to do cleaning up tasks, e.g. closing files etc.

Q. Write down the function of the following IO Exception classes:

EOFException, FileNotFoundException, InterruptedException, IOException.

Ans: EOFException: Signals that an end of

the file or end of the stream has been reached unexpectedly during input.

FileNotFoundException: Informs that a file could not be found.

InterruptedException: Warns that an IO operation has been interrupted.

IOException: Signals the an IO exception of some sort has occurs.

Q. What are wrapper classes?

Ans: Wrapper classes are the part or Java's standard library java.lang and these convert primitive datatypes into an object. to be more specific, a wrapper class wraps a value of primitive types in an object.

Java provides the following wrapper classes: Boolean Integer, Float, Double, Character etc.

Q. Why do we need a wrapper class?

Ans: A wrapper class is needed to store primitive values in objects as well as in conversion from string to primitive type.

Q. Distinguish between data type and wrapper class.

Ans: A data type starts with lowercase letter and wrapper class starts with uppercase letter.

Q. Define String?

Ans: A string is a set of two or more than two characters, a set of characters with the digit or a statement written with in double quotes. e.g. “Happy New Year”, “Computer Application” etc.

Q. What is String Buffer? How we create a String Buffer?

Ans: String Buffer is a type of memory location, which allows reasonable space to contain a string such a way that any change brought affect the same string.

String Buffer is created as follows:

```
StringBuffer p=new
```

```
StringBuffer(“Computer”);
```

Q. Differentiate between String and StringBuffer objects.

Ans: The String object of Java is immutable, i.e. once created they can not be changed. if any change occurs in a String object, then original object string remains unchanged and a new String is created with the changed String.

StringBuffer objects are mutable, on the other hand. That is these objects can be manipulated and modified as desired.

Q. Write down the purpose of the following string functions: toLowerCase(), toUpperCase(), replace(), trim(), equals(), length(), charAt(), concat(), substring(), indexOf(), compareTo().

Ans: The purpose and syntax of the following string functions are:

toLowerCase(): This function converts all the characters of the string in lower case.

for example:

```
String n="AMITABH";
```

```
n=n.toLowerCase();
```

```
System.out.println(n);
```

toUpperCase(): This function converts all

the characters of the string in upper case.

for example:

```
String n="amitabh";  
n=n.toUpperCase();  
System.out.println(n);
```

replace(): This function replace all the occurrence of a characters with another one.

```
String n="DAD";  
n=n.replace('D','G');  
System.out.println(n);
```

Trim(): This function is used to remove all the white spaces at the beginning and end of string.

```
String n="AMIT  “;  
n=n.trim();  
System.out.println(n);
```

equals(): This function is used to compare two string and give true or false if they are equal.

```
String s1="AMIT";  
String s2="amit";  
System.out.print(s1.equals(s2));
```

Length(): This function return the length characters present in the string.

```
String s="AMITABH";
```

```
System.out.print(s.length());
```

charAt(): This function return the nth character of the string.

```
String s="AMITABH";
```

```
System.out.print(s.charAt(2));
```

concat(): This function concatenate/join two strings.

```
String s1="AMITABH ";
```

```
String s2="BANERJEE"
```

```
System.out.print(s1.concat(s2));
```

substring(): This function returns the substring starting from the nth character of the string.

```
String s="AMITABH";
```

```
System.out.print(s.substring(3));
```

This function also returns the substring starting from the mth character upto the nth character without including the nth character of the string.

```
String s="AMITABH";
```

```
System.out.print(s.substring(2,4));
```

`indexOf()`: This function returns the position of the first occurrence a character in the string.

```
String s="AMITABH";
```

```
System.out.print(s.indexOf('A'));
```

This function also returns the position of the character from the nth position of the string.

```
String s="AMITABH";
```

```
System.out.print(s.indexOf('A',2));
```

`compareTo()`: This function returns negative if first string is less than second string, positive if greater and zero if equals.

```
String s1="AMIT";
```

```
String s2="SUMIT"
```

```
System.out.print(s1.compareTo(s2));
```

Q. What is the difference between `equals()` and `equalsIgnoreCase()` string functions?

Ans: Both the functions is used to compare strings, the difference being that `equals()` distinguishes between upper case and

lower case version of a character, where as equalsIgnoreCase() carries out comparison ignoring the case of characters.

Q. Differentiate between equals() and compareTo() methods.

Ans: Both the functions is used to comparing two strings, the difference being that (i) equals() method only comparing two string and gives they are equal or not, where as compareTo() methods also gives whether first string is greater or smaller then second one. (ii) equals() methods returns a boolean value, where as compareTo() methods return integer value.

Q. Differentiate between toLowerCase() and toUpperCase() methods.

Ans: The given two string method's change the case of the current string. The toLowerCase() method change the current string object to its equivalent Lower Case, where as toUpperCase() method change

the current string object to its equivalent Upper Case.

Q. What is the difference between the length() and capacity() string function.

Ans: The function length() returns the number of character contains in a string. Where as capacity() returns the maximum number of character that can be stored in a string objects.

Q. Name some of the most used packages?

Ans: Language extensions java.lang, utilities java.util, input-output java.io, GUI java.awt and javax.applet, network services java.net etc.

Q. Define static members?

Ans: The members that are declared static are called static members. These members are associate with the class it self rather than individual objects.

Q. What are static variables?

Ans: Static variables are used when we

want to have a variable common to all instances of a class.

Q. What are the restrictions of static methods?

Ans: (i) They can only call other static methods.

(ii) They can only access static data.

(iii) They cannot refer to 'this' or 'super' keywords in anyway.

Q. What are packages?

Ans: Java contains extensive library of pre-written classes we can use in our programs. These classes are divided into groups called packages. Various packages in Java are: java.applet, java.awt, java.io, java.lang, java.new, java.util etc.

Q. What are the benefits of organizing classes into packages.

Ans: In packages classes can be unique compared to other programs and be easily be reused.

Q. What are Java API packages:

Ans: Java API packages provide a large number of class grouped into different packages according to functionality.

Q. What are system packages?

Ans: The packages which are organised in hierarchical structure are referred as system packages.

Q. Explain the method on importing a package member?

Ans: To import a member of package into the current file, put an import statement at the beginning of the file before any class definitions but after the package statement, if there is one .

Q. Describe the method to import entire package?

Ans: To import a member all the classes contained in a particular package, using the import statement with the asterisk(*) wild card character.

Q. Distinguish between Static variable (class variable) and member variable (instance variable)

(i) Declare with the static keyword.

(ii) Exist at class level and can be used even if no instance of class exist in memory.

(iii) Created when class is first referred to.

(iv) Destroyed when the program is over.

(v) Can be accessed using either the class name or name of any instance of the class. (i) Declare without the static keyword.

(ii) Exist at instance level i.e. can not be used if there are no instance of class exist in memory.

(iii) Created with each instance.

(iv) Destroyed when the instance containing them is destroyed.

(v) Can be accessed using the name of the instance only to which they belong.

Q. Explain instance variable. Give an Example.

Ans: A data member that is created for every objects of the class.

```
public class abc
{
int a,b;    // instance variable or data
member
}
```

Q. State the difference between == operator and equals() method.

Ans: ==: 1. It is a relational operator. 2. it tests the value on the right side with value on the left side. equals(): 1. It is a string function. 2. It compares two strings and gives the value as true or false.