DEPARTMENT OF INFORMATION TECHNOLOGY

Java Programming (IT-506)

First List of practical

1) Write a program in java to find the factorial of a given number (While loop).
2) Write a program in java to find the largest among three numbers(if-else statement).
3) Write a program in java to swap two numbers without using auxiliary memory location (Using Bitwise XOR operation).
4) Write a program in java to implement linear search using enhanced for loop.
5) Write a program in java to find sum of all elements stored in an array. (enhanced for loop)
6) Write a program in java to demonstrate that iteration variable is read only in nature.
7) Write a program in java to implement Bubble sort algorithm. (Nested traditional for loop).
8) Write a program to implement Binary search algorithm.
9) Write a program in java to implement conditional operator.
10) Write a program in java to demonstrate all Bitwise operators.
11) Write a program in java to demonstrate short-circuit Boolean operators.
12) Demontrate the use of labelled break and labelled continue.
13) Using class and objects create a program to calculate the volume of a three dimensional object.
14) Add default constructor and overloaded parametrized constructors to

automate the initialization of the objects of your class.

15) Demonstrate the use of this keyword.
16) Write a program in java to create a class called Complex that defines the attribute (real & imaginary)and behaviour (add,mul,sub,comparison) of Complex numbers.
17) Using call by value, swap two numbers (in Complex Class).
18) Using call by reference, swap two numbers(in Complex Class).
19) Write a program in java to find factorial of a number using recursion.
20) Write a program to print Fibonacci series using recursion.
21) Write a program to print an array using recursion.
22) Write a program to implement Quick Sort.
23) Write a program to implement Merge Sort.
24) Write a program to implement Tower of Hanoi.
25) Find the smaller of two numbers to demonstrate static data members and static methods (in same class as of main and in a separate class).
26) Demonstrate static blocks.
27) Write a program in java to create Array of Arrays with rows of different length. Use length attribute of array object to print the multi-D array.
28) Write a program to demonstrate Static nested classes.
29) Write a program to demonstrate inner classes in java.
30) Write a program to print command line arguments.
31) Write a program to add numbers command line arguments.
32) Write a program to create a method that finds average of variable number of arguments using variable length args.
33) Write a program to overload varargs method.

34) Write a program to implement Single inheritance in java.
35) Write a program to implement Single inheritance with super to call Superclass constructors.
36) Demonstrate the use of super to resolve instance variable name hiding.
37) Demonstrate Member access modifier as it applies to subclass and its superclass.
38) Write a program to implement Multi-Level inheritance.
39) Demonstrate the sequence of call of constructors in mutli-level inheritance.
40) Using run-time polymorphism (Hierarchical Inheritance,overriding & Dynamic dispatch) find area of 2-D figures with figure as the base class.
41) Use abstract class to implement run-time polymorphic version of above program more effectively.
42) Demonstrate the use of final keywork as it applies to creation of constants, preventing overriding and preventing inheritance.
43) Write a program to demonstrate creation and use of packages and sub-packages (in separate programs).
44) Write a program to demonstrate Access protection (private, default modifier, protected,public) as it applies to classes,subclasses and packages in java.
45) Demonstrate the use of import statement and static import statement to calculate Hypotenuse of a right triangle(in separate programs).
46) Write a program to implement a stack using interfaces.
47) Write a program to demonstrate inheritance of two or more interfaces

by another interface and then its implementation by a class.

48) Write individual programs to handle following Exception using try and catch block :
    a) ArithmeticException
    b) ArrayIndexOutofBoundsException
    c) NegativeArraySizeException
    d) ArrayStoreException
    e) ClassCastException
    d) IllegalArgumentException
    e) NumberFormatException
49) Handle some of the exceptions written above using Multiple catch clauses in a single program (ensure each of them is thrown based on some condition).
50) Demonstrate throw keyword use to throw an exception explicitly and its handling.
51) Demonstrate throws keyword use to handle checked exceptions like
    a) ClassNotFoundException
    b) CloneNotSupportedException
52) Demonstrate try-catch-finally blocks with all possible variations.
53) Demonstrate the creation of user-defined exception classes in java.
54) Use initCause and getCause methods of Throwable to demonstrate chained exception in java.
55) Write a program to get reference to the currently executing thread and then to rename and print it .
56) Write a program to create two threads of execution by implementing runnable interface.

57) Write a program to extend Thread class to create two threads of execution.
58) Write a program to demonstrate the difference between a deamon thread and a user thread.
59) Write a program to create three or more threads of control.
60) Demonstrate the use of join( ) method in Thread class.
61) Write a program to demonstrate priority based thread execution.
62) Create a dummy bank account,withdraw and deposite to demonstrate the race condition due to unsynchronized threads.
63) Synchronize the above application using Synchronization methods.
64) Synchronize the above application using Synchronization stmt.
65) Write a program in java to provide solution to a single buffer producer-consumer problem(Inter Process Communication).
66) Write a program in java to provide a solution to n(=10) buffer size producer-consumer problem.
67) Write a program in java to demonstrate deadlock in threads.
68) WAP to demonstrate enumerations as class types in Java.
69) WAP to demonstrate all the operations on Integer Wrapper class.
70) WAP to demonstrate all the operations on Float Wrapper class.
71) WAP to demonstrate all the operations on Boolean Wrapper class.
72) WAP to demonstrate all the operations on Character Wrapper class.
73) WAP to demonstrate Autoboxing & Auto-unboxing capabilities of Java as applicable to assignments,method call & expressions.
74) WAP to demonstrate various ways of creating a String Object.
75) WAP to demonstrate following operations on Strings:

- Initialization of string using literals.

- Concatenation of strings.
- Explicit data conversion using valueOf().
- Finding out String Length.

76) WAP to demonstrate String Extraction methods in String class.

77) WAP to demonstrate all String comparison methods.

78) WAP to demonstrate String Search methods in **String.**

79) WAP to demonstrate subString(),concat(),replace() and trim(),toLowerCase() and toUpperCase().

80) WAP to demonstrate the following methods in **StringBuffer:**

    I) length() and capacity()

    II) ensureCapacity() ,setLength()

    III) chatAt() and setCharAt()

    IV) getChars() , append( ),insert( ), reverse( ), delete()

    V) replace( ) and Substring( )

81) WAP to read a single byte, an array of bytes and a subrange of bytes from the standard input stream attached to console.

82) WAP to write a single byte, an array of bytes and a subrange of bytes to the standard output stream linked to console.

83) WAP to program to print all system properties using getProperties() of System class.

84) Use currentTimeMillis() to compute time take by a program to execute.

85) WAP to demonstrate testing & checking methods on file objects.

86) WAP to demonstrate querying methods on existing files or directories.

87) WAP to enlist the contents of a directory ending with a specific file

extension using FilenameFilter interface.

88) WAP to demonstrate all methods available in File class to create and rename a file, to create a directory or a hierarchy of directory.
89) WAP to demonstrate all FileInputStream methods to read bytes from a files.
90) WAP to demonstrate all FileOutputStream methods to write bytes to a files.
91) WAP to copy one file to another(Irrespective of extension) using BufferedInputStream & BufferedOutputStream.
92) WAP to demonstrate ByteArrayInputStream.
93) WAP to demonstrate ByteArrayOutputStream.
94) WAP to demonstrate all the methods defined in PushbackInput stream class.
95) WAP to demonstrate SequenceInputStream.
96) WAP to demonstrate all the methods in RandomAccessFile class.
97) WAP to write and read primitive types to and from a byte stream using DataOutputStream & DataInputStream resp.
98) WAP to demonstrate various aspects of Formatter class for precisely formatting the data values.
99) WAP to demonstrate all the format flags available in formatter class for precise formatting of data
100) WAP to redirect standard output stream to a file using print stream.
101) WAP to demonstrate reading content of a file using File Reader.
102) WAP to demonstrate writing characters to a files using FileWriter class.
103) WAP to demonstrate CharArrayReader.

104) WAP to demonstrate CharArrayWriter.

105) WAP to demonstrate PushBackReader.

106) WAP to demonstrate all the constructor and methods available in PrintWriter class.

107) WAP to read sensible input from keyboard using stream tokenizer.

108) WAP to add two numbers stored in a file using StreamTokenizer class.

109) WAP to add two numbers by reading them from console & give five number of chances to use for each value in case of wrong input.

110) WAP to count the number of words ,char & lines without using capabilities of StreamTokenizer.

111) WAP to count the no. of words ,char &lines using Stream Tokenizer

112) WAP to customize the StreamTokenizer class to recognize & print only Hindi alphabets.

113) WAP to print first 5000 unicode character in a file.

114) WAP to serialize & deserialize an object of class that contains three instance variable of primitive types & overrides toString () methods.

115) WAP to demonstrate **transient** keyword.

116) WAP to demonstrate the importance of default constructor of a base class when the base class is not serialized.

117) WAP to serialize an object of the a derived class BoxShipment extended from BoxHeight inherited from Box class.

First Date of Submission: 12 November 2011, Saturday.