NIT, KURUKSHETRA

General Instruction for Laboratory Classes

DO'S

Without prior permission do not enter into the laboratory.
Students should come with observation and record notebook to the laboratory.
Students should maintain silence inside the laboratory.
After completing the laboratory exercise, make sure to shutdown the system properly.

DONT'S

Students bringing the bags inside the laboratory.
Students using the computers in an improper way.
Students scribbling on the desk and mishandling the chairs.
Students using mobile phones inside the laboratory.

Students making noise in the laboratory

Lab Manual 1

Aim: - Introduction to Java.

Java was conceived by James Gosling and his team mates at Sun Microsystems, Inc. in 1991.It took 18 months to develop the first working version. This language was initially called “Oak”, but was renamed “Java” in 1995.
Java is pure object oriented language which fully utilized the OOPS concept like Encapsulation, Abstraction, Inheritance, Polymorphism. The main feature of JAVA language is that, it is secured and *platform independent* (architecture neutral) language as, the bytecode (highly optimized set of instructions designed to be executed by the Java run-time system) can run under any environment independent of hardware and programming constraints.

To download and install latest version of java

Java Development Kit (JDK), officially named "Java Platform Standard Edition (Java SE)", which is freely available from Sun Microsystems (now part of Oracle), is needed for writing Java programs.
The mother site for JDK (Java SE) is http://www.oracle.com/technetwork/java/index.html.
JRE (Java Runtime) is needed for *running* Java programs. JDK (Java Development Kit), which includes JRE plus the development tools (such as compiler and debugger), is need for *writing* as well as *running* Java programs. Since you are supposed to write Java Programs, you should install JDK, which includes JRE.

To install jdk do follow the instructions given in the link below:-
https://www3.ntu.edu.sg/home/ehchua/programming/howto/JDK_Howto.html

Program1: WAP to print the Hello World.
```
class Example
{
        public static void main(String args[])
        {
                System.out.println("Hello World”);
        }
}
```

Program2: WAP to instantiate a variable and printing its value.
```
class JavaFun
{
    public static void main(String args[])
    {
        int num;
        num = 100;
        System.out.println("This is num: " + num);
```

```
            num = num * 2;
            System.out.print("The value of num * 2 is ");
            System.out.println(num);
            }
}
```

Program3: WAP to print numbers from 1 to 10

```
class ForTest
{
      public static void main(String args[])
      {
             int x;
            for(x = 0; x<10;x=x+1)
            System.out.println("This is x: "+x);
      }
}
```

Program4: Take a range from 1-50 and write a program that tells whether a number is odd or even.

Program5: WAP to display the output as below:

```
*
**
***
****
*****
```

Program6: WAP to add two integers and value is to be initiated at compile time.

Lab Manual 2

Aim: To study about the various data primitive types.

There are eight primitive datatypes supported by Java. Primitive datatypes are predefined by the language and named by a keyword.

byte

- Byte data type is an 8-bit signed two's complement integer
- Minimum value is -128 (-2^7)
- Maximum value is 127 (inclusive)(2^7 -1)
- Default value is 0
- Byte data type is used to save space in large arrays, mainly in place of integers, since a byte is four times smaller than an integer.
- Example: byte a = 100, byte b = -50

short

- Short data type is a 16-bit signed two's complement integer
- Minimum value is -32,768 (-2^15)
- Maximum value is 32,767 (inclusive) (2^15 -1)
- Short data type can also be used to save memory as byte data type. A short is 2 times smaller than an integer
- Default value is 0.
- Example: short s = 10000, short r = -20000

int

- Int data type is a 32-bit signed two's complement integer.
- Minimum value is - 2,147,483,648 (-2^31)
- Maximum value is 2,147,483,647(inclusive) (2^31 -1)
- Integer is generally used as the default data type for integral values unless there is a concern about memory.
- The default value is 0
- Example: int a = 100000, int b = -200000

long

- Long data type is a 64-bit signed two's complement integer
- Minimum value is -9,223,372,036,854,775,808(-2^63)
- Maximum value is 9,223,372,036,854,775,807 (inclusive)(2^63 -1)
- This type is used when a wider range than int is needed
- Default value is 0L
- Example: long a = 100000L, long b = -200000L

float

- Float data type is a single-precision 32-bit IEEE 754 floating point
- Float is mainly used to save memory in large arrays of floating point numbers
- Default value is 0.0f
- Float data type is never used for precise values such as currency
- Example: float f1 = 234.5f

double

- double data type is a double-precision 64-bit IEEE 754 floating point

- This data type is generally used as the default data type for decimal values, generally the default choice
- Double data type should never be used for precise values such as currency
- Default value is 0.0d
- Example: double d1 = 123.4

boolean

- boolean data type represents one bit of information
- There are only two possible values: true and false
- This data type is used for simple flags that track true/false conditions
- Default value is false
- Example: boolean one = true

char

- char data type is a single 16-bit Unicode character
- Minimum value is '\u0000' (or 0)
- Maximum value is '\uffff' (or 65,535 inclusive)
- Char data type is used to store any character
- Example: char letterA = 'A'

Program1: Run the following program and find out its output.

```
class Test
{
        public static void main(String args[])
         {
                int x, y;
                y = 20;
                for(x = 0; x<10; x++)
                {
                        System.out.println("This is x: "+x);
                        System.out.println("This is y: "+y);
                        y=y-2;
                }
        }
}
```

Program2: WAP to find the distance travel by light. Consider the speed of light as 18600 miles/ second and number of days=1000.

Program3: WAP to compute the area of circle. Given that the radius of the circle is 10.8.

Program4: WAP to find out the ASCII value of given character.

Program5: WAP to find out the character of given ASCII value.

Program6: WAP to add two no's entered through command line.

Program7: Write a Program in java to read the 5 integers and sort them in ascending order.

Program8: Write a simple program to take 3 inputs by user and calculate the greatest no in all.

Program9: WAP to determine the sum of the following harmonic series for a given value of n: 1+1/2+1/3+1/4…………1/n the value of n should be given interactively through the keyboard.

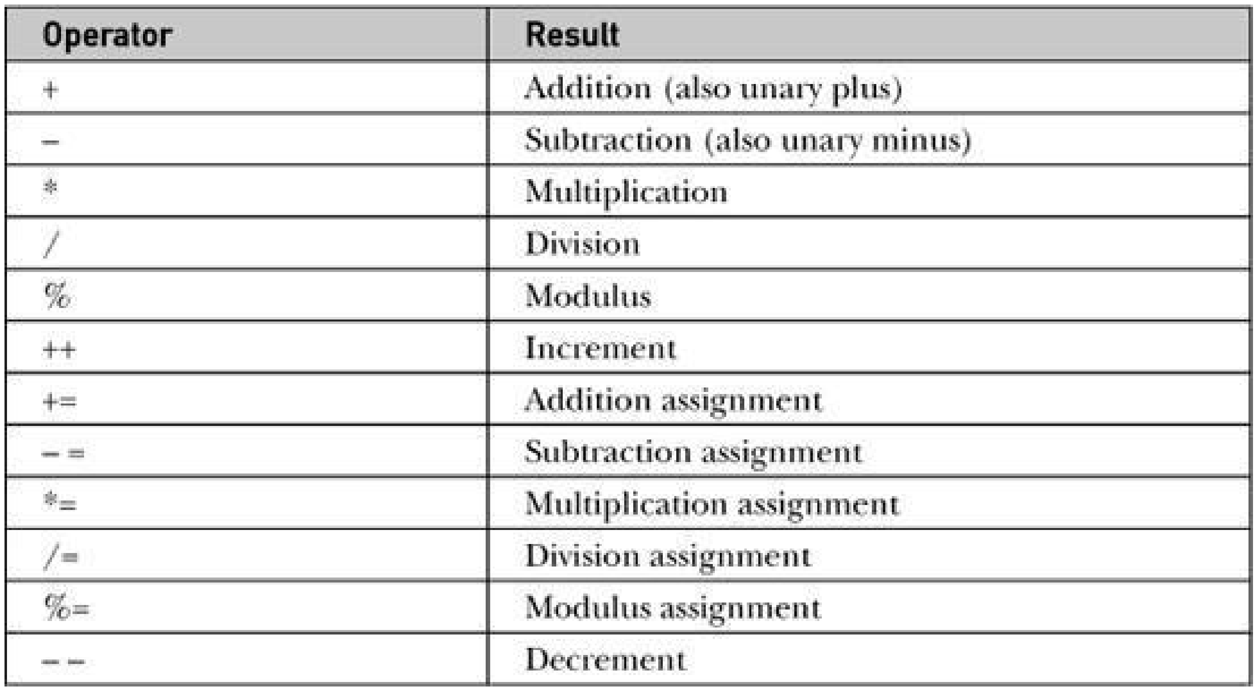
Lab Manual 3

Aim: To study about constants, variables, operators & expressions, and arrays in java.

A **variable** provides us with named storage that our programs can manipulate. Each variable in Java has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.
Basic form of a variable declaration is shown here:
type identifier [= value][, identifier [= value] ...] ;
The type is one of Java's atomic types, or the name of a class or interface. The identifier is the name of the variable.

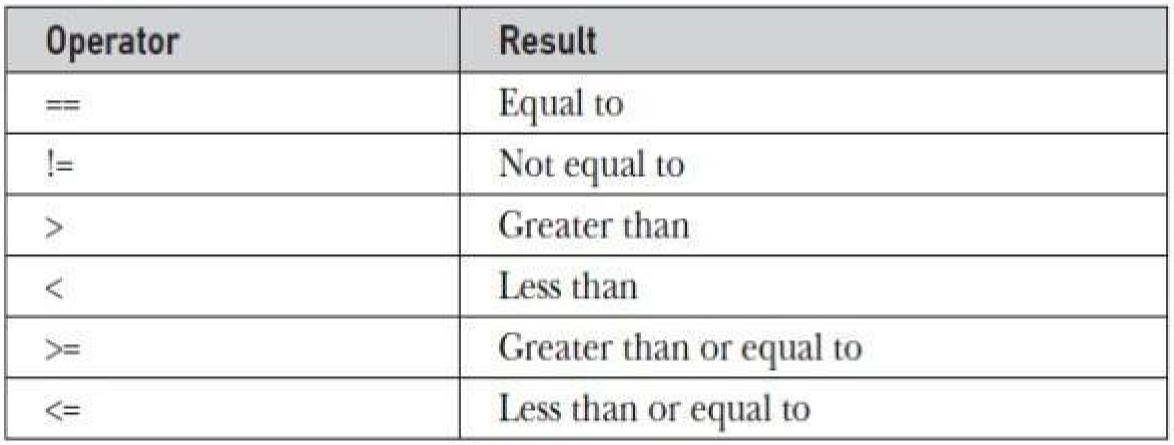
All the Java **operators** into the following groups –

Arithmetic Operators: - Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra.



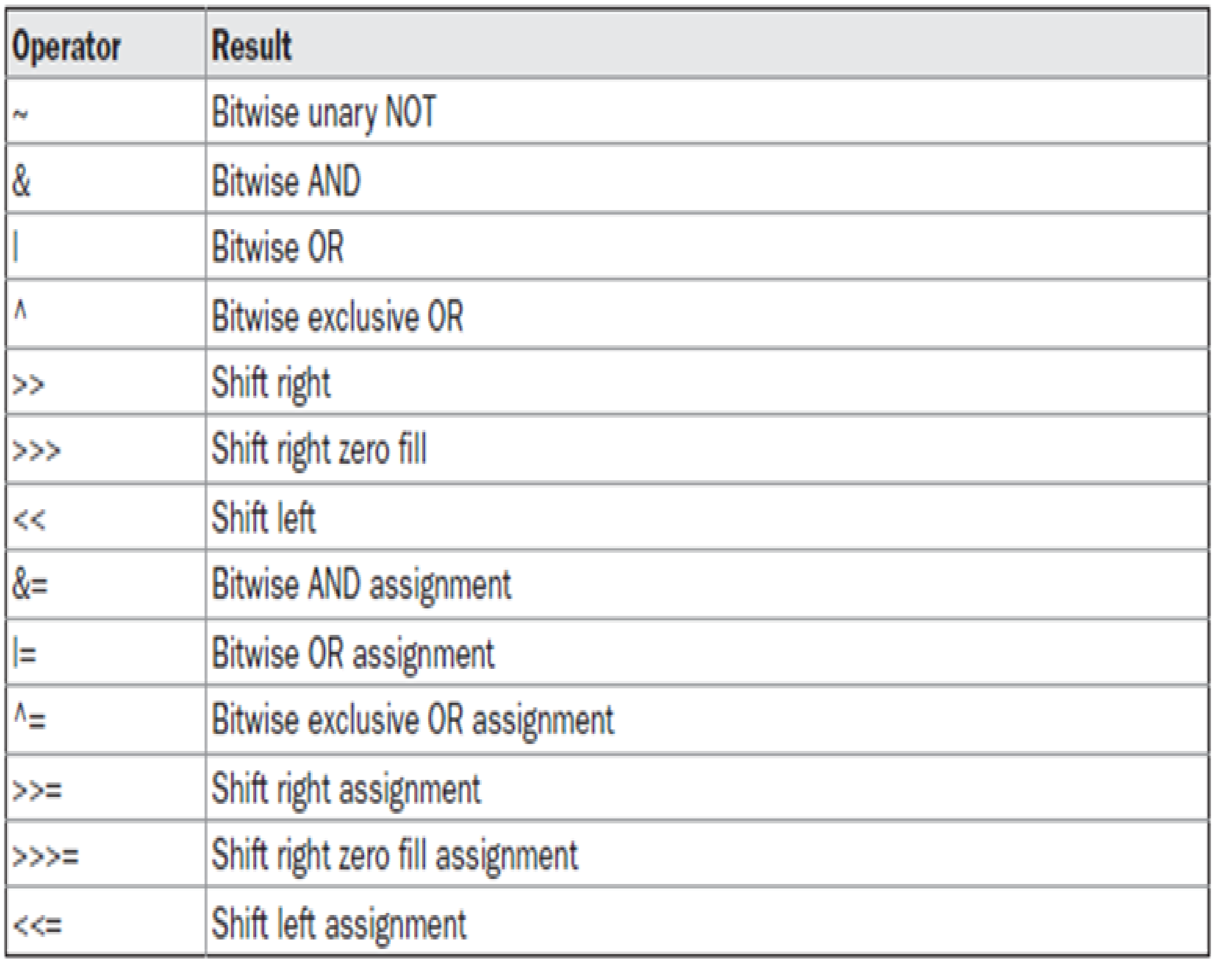
Operator	Result
+	Addition (also unary plus)
–	Subtraction (also unary minus)
*	Multiplication
/	Division
%	Modulus
++	Increment
+=	Addition assignment
– =	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
– –	Decrement

Relational Operators: - There are following relational operators supported by Java language.



Operator	Result
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Bitwise Operators: - Java defines several bitwise operators, which can be applied to the integer types, long, int, short, char, and byte.



Operator	Result
~	Bitwise unary NOT
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
>>	Shift right
>>>	Shift right zero fill
<<	Shift left
&=	Bitwise AND assignment
|=	Bitwise OR assignment
^=	Bitwise exclusive OR assignment
>>=	Shift right assignment
>>>=	Shift right zero fill assignment
<<=	Shift left assignment

Logical Operators: - The Boolean logical operators shown here operate only on boolean operands. All of the binary logical operators combine two boolean values to form a resultant boolean value.

Operator	Result
&	Logical AND
|	Logical OR
^	Logical XOR (exclusive OR)
||	Short-circuit OR
&&	Short-circuit AND
!	Logical unary NOT
&=	AND assignment
|=	OR assignment
^=	XOR assignment
==	Equal to
!=	Not equal to
?:	Ternary if-then-else

An **array** is a group of like-typed variables that are referred to by a common name. Arrays of any type can be created and may have one or more dimensions.
A one-dimensional array is, essentially, a list of like-typed variables. The general form of a one-dimensional array declaration is
type var-name[ ];
In Java, multidimensional arrays are actually arrays of arrays. To declare a multidimensional array variable, specify each additional index using another set of square brackets.

Program1: What will be the result of compiling and running the following program:-

(a) class CharDemo

```
{
        public static void main(String args[])
        {
                char ch1, ch2; ch1 = 88; // code for X
                ch2 = 'Y';
                System.out.print("ch1 and ch2: ");
                System.out.println(ch1 + " " + ch2);
         }
}
```

(b) class BoolTest

```
{
        public static void main(String args[])
        {
                boolean b;
                b = false;
                System.out.println("b is " + b);
                b = true;
                System.out.println("b is " + b);
                if(b) System.out.println("This is executed.");
                b = false; if(b) System.out.println("This is not executed.");
                System.out.println("10 > 9 is " + (10 > 9));
        }
}
```

Program2: WAP in which you declare a character type variable. Assign it a certain char type value and the post increment it and print its value.

Program3: WAP to calculate the hypotenuse of a triangle. (Hint: Math. sqrt(var_name/value)

Program4: Write a program which sorts an array of type integer.

Program5: WAP to convert the given temperature in Fahrenheit to Celsius using the following conversion formula C=(F*32)/1.8.

Program6: WAP to find all the numbers and sum of all integers greater than 100 less than 200 that are divisible by 7.

Program7: Write a program in java to read 10 floating point numbers in an array and display the numbers in reverse order.

Program8: WAP to have the following output:

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0

0 0 0 0 0
0 1 2 3 4
0 2 4 6 8
0 3 6 9 12

0 0 0 0 0
0 2 4 6 8
0 4 8 12 16
0 6 12 18 24
```

Lab Manual 4

Aim: To study about control structures, classes, objects & methods in Java.

A **class** declaration names the class and encloses the class body between braces. The class name can be preceded by modifiers. The class body contains fields, methods, and constructors for the class. A class uses fields to contain **state** information and uses methods to implement **behavior.** Constructors that initialize a new instance of a class use the name of the class and look like methods without a return type.

You control access to classes and members in the same way: by using an access modifier such as `public` in their declaration.

You create an object from a class by using the new operator and a constructor. The new operator returns a reference to the object that was created. You can assign the reference to a variable or use it directly.

Instance variables and methods that are accessible to code outside of the class that they are declared in can be referred to by using a qualified name. The qualified name of an instance variable looks like this:

objectReference.variableName

The qualified name of a method looks like this:

objectReference.methodName(argumentList)

Program1: Discover what error messages the compiler produces when you make each of the following mistakes. How many unique error messages are you able to cause the compiler to produce?

1. **Naming your file incorrectly, then compiling.**
2. **Forgetting a keyword such as void or class**
3. **Forgetting a quotation mark "**
4. **Forgetting a parenthesis (or)**
5. **Forgetting a dot .**
6. **Using too many or too few braces { or }**

Program 2: What output is produced by the following Java method?

```
        public static void stars()
{
                for(int i=1; i<=10; i++)
                {
                        for (int j = 1;  j <= i;  j++)
                {
                        System.out.print("*");

                        }
                        for (int j = 1; j <= 20 - 2 * i; j++)
                        {
                                System.out.print(" ");
                        }
                        for(int j=1;j<=i;j++)
                        {
                                System.out.print("*");
                        }
                        System.out.print("*");
                        }
        }
}
```

Program3: Write a method named numUnique that accepts three integers as parameters and that returns the number of unique integers among the three. For example, the call numUnique(18, 3, 4) should return 3 because the parameters have 3 different values. By contrast, the call numUnique(6, 7, 6)would return 2 because there are only 2 unique numbers among the three parameters: 6 and 7.

Program4: Given a list of marks ranging from 0 to 100, write a program to compute and print the numbers of student should have obtained marks (a) in the range 81 to 100 (b) in the range 61 to 80 (c) in the range 41 to 60 (d) in the range 0 to 40. The program should use minimum number of if statements.

Program5: Admission to a professional course is subject to the following conditions:

Marks in mathematics>=60
Marks in physics>=50

Marks in chemistry>=40
Total in all subjects >= 200 (or)
Total in mathematics and physics>=150 given the marks in the 3 subjects.

Write a program to process the application to list the eligible candidates.

Program6: Create a class BOX which has default constructor and parameterized constructor to initialize the values. Calculate the volume of BOX. Implement the use of "this" reference in this program.

Program7: Create a class named "Addition" having the following overload methods.

Int add(int x, int y)
Double add(double x, double y)
String add(String x, String y)
Int add([] x)

Program8: Define a class to represent a bank account. Include the following members:
Data Members

1. **Name of Depositors**
2. **Account Number**
3. **Type of Account**
4. **Balance amount in the Account**

Methods

1. **To assign initial values**
2. **To deposit an amount**
3. **To withdraw an amount after checking the balance**
4. **To display name and balance**

Program9: Create a java class which has method to find out the factorials of numbers from 1 to 10 using recursion.

Lab Manual 5

Aim: To study about inheritance & packages in Java.

In the Java language, classes can be derived from other classes, thereby inheriting fields and methods from those classes.

A class that is derived from another class is called a subclass (also a derived class, extended class, or child class). The class from which the subclass is derived is called a superclass (also a base class or a parent class).

Except for the Object class, a class has exactly one direct superclass. A class inherits fields and methods from all its superclasses, whether direct or indirect. A subclass can override methods that it inherits, or it can hide fields or methods that it inherits. (Note that hiding fields is generally bad programming practice.)

The Overriding and Hiding Methods shows the effect of declaring a method with the same signature as a method in the superclass.

The Object class is the top of the class hierarchy. All classes are descendants from this class and inherit methods from it.

You can prevent a class from being subclassed by using the final keyword in the class's declaration. Similarly, you can prevent a method from being overridden by subclasses by declaring it as a final method.

An abstract class can only be subclassed; it cannot be instantiated. An abstract class can contain abstract methods—methods that are declared but not implemented. Subclasses then provide the implementations for the abstract methods.

Java provides a mechanism for partitioning the class name space into more manageable chunks. This mechanism is the package. The package is both a naming and a visibility control mechanism

Program1: What will be the output of the below program?

```
class A
{
   {
      System.out.println(1);
   }
}

class B extends A
{
   {
      System.out.println(2);
   }
}

class C extends B
{
   {
      System.out.println(3);
   }
}

public class MainClass
{
   public static void main(String[] args)
   {
      C c = new C();
   }
}
```

Program2: **Below code is showing compile time error. Can you suggest the corrections?**

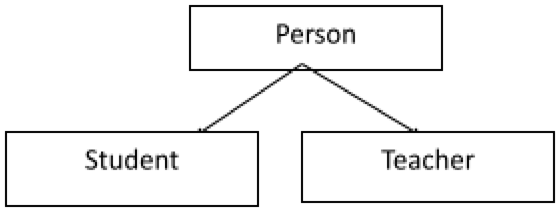
```
class X
{
   public X(int i)
   {
      System.out.println(1);
   }
}

class Y extends X
{
   public Y()
   {
      System.out.println(2);
   }
}
```

Program3: Write a program to identify the accessibility of a variable by means of different access specifiers within and outside package.

Program4: Write an employee class Marketer to accompany the other employees. Marketers make $50,000 ($10,000 more than general employees), and they have an additional method named advertise that prints "Act now, while supplies last!" Use the super keyword to interact with the Employee superclass as appropriate.

Program5: Create a Base Class Person and two derived class as student and teacher with their constructor and method. Assume the student to be in the same package as that of person and teacher class to be in different package.
The inheritance hierarchy would appear as follows:



1. Add methods "get" the instance variables in the Person class. These would consist of: getName, getAge, getGender.
2. Add methods to "set" and "get" the instance variables in the Student class. These would consist of: getIdNum, getGPA, setIdNum.
3. Write a Teacher class that extends the parent class Person.

Program6: Describe abstract class called Shape which has three subclasses say Triangle, Rectangle, Circle. Define one method area() in the abstract class and override this area() in these three subclasses to calculate for specific object i.e. area() of Triangle subclass should calculate area of triangle etc. Same should be for Rectangle and Circle.

Program7: Assume that a bank maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.
Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:
(a) Accept deposit from a customer and update the balance.
(b) Display the balance.
(c) Compute and deposit interest.
(d) Permit withdrawal and update the balance.
(e) Check for the minimum balance, impose penalty, if necessary and update the balance.

Lab Manual 6

Aim: To study about Interfaces and Exception Handling in Java.

An interface in java is a blueprint of a class. It has static constants and abstract methods.
The interface in java is **a** mechanism to achieve abstraction**.** There can be only abstract methods in the java interface not method body. It is used to achieve abstraction and multiple inheritance in Java. Interface keyword is used to define an interface and for implementing the interface, implement keyword is used.

A program can use exceptions to indicate that an error occurred. To throw an exception, use the throw statement and provide it with an exception object — a descendant of Throwable — to provide information about the specific error that occurred. A method that throws an uncaught, checked exception must include a throws clause in its declaration.
A program can catch exceptions by using a combination of the try, catch, and finally blocks.
The try block identifies a block of code in which an exception can occur.
The catch block identifies a block of code, known as an exception handler that can handle a particular type of exception.
The finally block identifies a block of code that is guaranteed to execute, and is the right place to close files, recover resources, and otherwise clean up after the code enclosed in the try block.
The try statement should contain at least one catch block or a finally block and may have multiple catch blocks.

Program1: WAP to compute the area of circle and rectangle using Interfaces.

Program2: WAP that uses interface for the implementation of fixed-size and dynamic-size stacks (for dynamic stack size should be redefine as per the number of elements).

Program3: Is the following code legal?
try { } finally { }

Program4: Write a program for exception handling with StringIndexOutOfBound exception

- **Create an object of the class having StringIndexOutOfBound exception whenever an index is invoked of a string, which is not in the range.**
- **Each character of a string object is stored in a particular index starting from 0.**
- **To get a character present in a particular index of a string you can use a method charAt(int) of java.lang.String where int argument is the index.**

Program5: An array is declared with 5 elements. Then the code tries to access the 6th element of the array which throws an exception. Write the program for this.

Program6: Write a suitable program for the following conditions:
1. A try block followed by multiple catch blocks
2. Catching multiple type of exceptions
3. Using throws/throw keywords
4. Using finally block
5. Using try-with-resources
6. User-defined exceptions

Lab Manual 7

Aim: To study about Multi-Threading in java.

In Java, a Thread is essentially the Object that represents one piece of work. When you start your application and it starts to run, Java has created a Thread and this Thread is what will carry out the work that your application is meant to do.

Java multithreading allows you to do multiple tasks at the same time**.**

Commonly used methods of Thread class:
1. **public void run():** is used to perform action for a thread.
2. **public void start():** starts the execution of the thread.JVM calls the run() method on the thread.
3. **public void sleep(long miliseconds):** Causes the currently executing thread to sleep (temporarily cease execution) for the specified number of milliseconds.
4. **public void join():** waits for a thread to die.
5. **public void join(long miliseconds):** waits for a thread to die for the specified miliseconds.
6. **public int getPriority():** returns the priority of the thread.
7. **public int setPriority(int priority):** changes the priority of the thread.
8. **public String getName():** returns the name of the thread.
9. **public void setName(String name):** changes the name of the thread.
10. **public Thread currentThread():** returns the reference of currently executing thread.
11. **public int getId():** returns the id of the thread.
12. **public Thread.State getState():** returns the state of the thread.
13. **public boolean isAlive():** tests if the thread is alive.
14. **public void yield():** causes the currently executing thread object to temporarily pause and allow other threads to execute.
15. **public void suspend():** is used to suspend the thread(depricated).
16. **public void resume():** is used to resume the suspended thread(depricated).
17. **public void stop():** is used to stop the thread(depricated).
18. **public boolean isDaemon():** tests if the thread is a daemon thread.
19. **public void setDaemon(boolean b):** marks the thread as daemon or user thread.
20. **public void interrupt():** interrupts the thread.
21. **public boolean isInterrupted():** tests if the thread has been interrupted.

Program1:

Suppose there are 5 workers and carry out a work at the same time. Develop a program which shows this scenario with the concept of multithreading.
Hint.
1. Create 5 workers with the help of for loop
2. Use Thread.sleep() method that pauses the thread for a custom defined period of time. When we pause a Thread, this would simulate that Thread being busy doing some sort of actual work.
3. Suppose 5 workers works for 5 seconds each , then a total of 5 seconds would be the total time for all 5 workers if we use the concept of multithreading .

Program2: How you can achieve the following output with the help of Sleep method in multithreading concept in java
Output

```
1
1
2
2
3
3
4
4
```

Program3: Write a java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds.

Program4: Assume the developer has created a thread class and that the main method is as follows:

```
public static void main(String [] args)
{
Thread th1 = new MyThread();
Thread th2 = new MyThread();
th1.run();
th2.run();
}
```

What will happen when executing this main method? Briefly describe the consequences.

Program5: Implement a class that checks whether a given number is a prime using both the Thread class and Runnable interface.

Program6: Write a java program that correctly implements producer consumer problem using the concept of inter thread communication.

Lab Manual 8

Aim: To study about Strings in java.

In java, string is basically an object that represents sequence of char values. An array of characters works same as java string. The CharSequence interface is used to represent sequence of characters. It is implemented by String, StringBuffer and StringBuilder classes. It means, we can create string in java by using these 3 classes. All these three classes are members of **java.lang** package and they are final classes. That means you can't create subclasses to these three classes. The java String is immutable i.e. it cannot be changed. Whenever we change any string, a new instance is created. For mutable string, you can use StringBuffer and StringBuilder classes. Only **String** and **StringBuffer** objects are thread safe. **StringBuilder** objects are not thread safe. So whenever you want immutable and thread safe string objects, use *java.lang.String* class and whenever you want mutable as well as thread safe string objects then use *java.lang.StringBuffer* class.

Program1: WAP to find whether a given string is palindrome or not.
Program2: Write a method that will remove given character from the String.
Program3: Write a java program for sorting a given list of names.
Program4: Write a program in java to read a sentence and count the words. For Example:

Enter here a Sentence: "This is a java Program".

Number of words: 5

Program5: Write a program that computes your initials from your full name and displays them.

Program6: An anagram is a word or a phrase made by transposing the letters of another word or phrase; for example, "parliament" is an anagram of "partial men," and "software" is an anagram of "swear oft." Write a program that figures out whether one string is an anagram of another string. The program should ignore white space and punctuation.

Lab Manual 9

Aim: To study about Applets in java.

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side. Advantages of Applets include the following:

It works at client side so less response time.

Secured

It can be executed by browsers running under many platforms, including Linux, Windows, Mac Os etc.

Disadvantages of Applets include:

Plugin is required at client browser to execute applet.

Applet class extends the AWT class Panel. Panel extends Container which is the subclass of Component which in-turn extends the Object class.

These classes provide support for Java's window-based, graphical interface. Thus, Applet provides all of the necessary support for window-based activities.

Program1: Write an applet to display a simple message on a colored background.

Program2: Write an applet to display a moving banner showing the status of it.

Program3: Write an applet to draw a simple and beautiful landscape.

Program4: WAP to play audio in applet with appropriate requirements of AWT components.

Program5: WAP to have communication between two applets.

Program6: WAP to display clock in an applet.

Lab Manual 10

Random programs

Program1: WAP to implement magic matrix in java. (A magic square of order n is an arrangement of n^2 numbers, usually distinct integers, in a square, such that the n numbers in all rows, all columns, and both diagonals sum to the same constant.)

Program2: WAP to implement N-Queen Problem. (The N Queen is the problem of placing N chess queens on an N×N chessboard so that no two queens attack each other.)

Program3: WAP to insert elements in n^2 matrix in spiral form.

1	2	3	4
12	13	14	5
11	16	15	6
10	9	8	7

Program4: WAP to implement SRTF(Shortest remaining time first) to compute average waiting time and average turn around time on input of number of user defined processes with their respective arrival times and burst time.

Program5: WAP to implement SCAN (disk scheduling) to compute total head movement. Input will be a disk request queue with user defined initial head position.