Modern Programming Languages

LAB MANUAL

**BSCS-5B (Morning)**

**Suleman Khurram**

# OBJECTIVES:

* To teach the students basics of JAVA programs and its execution.
* To teach the students the differences between C++ and Java programming.
* To make the students learn concepts like packages and interfaces.
* To make the students understand life cycle of the applets and its functionality.
* To make the students understand the usage util package.
* To teach the student, to develop java programs using interfaces.

# Recommended System/Software Requirements:

* Intel based desktop PC with minimum of 2.6GHZ or faster processor with at least 256 MB RAM and 40GB free disk space.
* Operating system: Flavor of any WINDOWS.
* Software:jdk1.7 to jdk16
* Linux and MvSQL.
* Eclipse or Netbeans.

# INTRODUCTION TO OOP

Object-oriented programming (OOP) is a computer science term used to characterize a programming language that began development in the 1960’s. The term ‘object-oriented programming’ was originally coined by Xerox PARC to designate a computer application that describes the methodology of using objects as the foundation for computation. By the 1980’s, OOP rose to prominence as the programming language of choice, exemplified by the success of C++. Currently, OOPs such as Java, J2EE, C++,C=, Visual Basic.NET, Python and java Script are popular OOP programming languages that any career-oriented Software Engineer or developer should be familiar with.

OOP is widely accepted as being far more flexible than other computer programming languages. OOPs use three basic concepts as the fundamentals for the Abstraction, Polymorphism, Event Handling and Encapsulation are also significant concepts within object- oriented programming languages that are explained in online tutorial describing the functionality of each concept in detail.

The java platform is undoubtedly fast moving and comprehensive. Its many application programming interfaces (APIs) provide a wealth of functionality for all aspects of application and system-level programming. Real-world developers never use one or two APIs to solve a problem, but bring together key functionality spanning a number of APIs, Knowing which APIs you need,

which parts of which APIs you need, and how the APIs work together to create the best solution can be a daunting task.

# 1. GUIDELINES TO STUDENTS

1. Equipment in the lab for the use of student community. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage caused is punishable.
2. Students are required to carry their observation book and lab records with completed exercises while entering the lab.
3. Lab records need to be submitted every week.
4. Students are not supposed to use pen drives in the lab.

# 2. LAB OBJECTIVE

* + To introduce Java compiler and eclipse platform.
  + To make the student learn an object oriented way of solving problems using java.
  + To make the students to write programs using multithreading concepts and handle exceptions.
  + To make the students to write programs that connects to a database and be able to perform various operations.
  + To make the students to create the Graphical User Interface using Applets, AWT Components & Swing Components.

# LAB OUTCOME

* + Able to use Java compiler and eclipse platform to write and execute java program.
  + Understand and Apply Object oriented features and Java concepts.
  + Able to apply the concept of multithreading and implement exception handling.
  + Able to access data from a Database with java program.
  + Develop applications using Console I/O and File I/O,GUI applications

# List of experiments as per the university curriculum

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Name of the Program** | **Done** |
| **1** | **Week1 :** | **Yes** |
| Run Netbeans IDE and create a simple HelloWorld program. Understand the basic structure of Java Program. |
| **2** | **Week 2 :** | **Yes** |
| Run and give outputs of following programs. |
| **3** | **Week 3 :** | **Yes** |
| Provide output of following programs |
| b)Write a program to print Fibonacci series of n terms where n is input by user :  0 1 1 2 3 5 8 13 24 ..... |
| **4.** | **Week 4 :** | **Yes** |
| Provide output of following code  public class ExceptionHandling  {  public static void main(String[] args)  {  try  {  int i = Integer.parseInt("abc"); //This statement throws NumberFormatException  }  catch(Exception ex)  {  System.out.println("This block handles all exception types");  }  catch(NumberFormatException ex)  {  System.out.println(“Number format exception occurred"); }  }  }  public class ExceptionHandling  {  public static void main(String[] args)  {  try  {  int i = Integer.parseInt("abc"); //This statement throws NumberFormatException  }  catch(Exception ex)  {  System.out.println("This block handles all exception types");  }  catch(NumberFormatException ex)  {  //Compile time error  //This block becomes unreachable as  //exception is already caught by above catch block  }  }  }  b) Try to write a program and throw an exception explicitly |
| **5** | **Week 5 :** | **Yes** |
| • Store the following string in a string variable:  “My university name is Arid, Agriculture University.”  Print all the tokens using StringTokenizer.  • Use String buffer to store “Welcome ”, then take the name as input from the user and append it to the string buffer and print to the console.  Output will be like:  Welcome Suleman |
| **6** | **Week 6 :** | **Yes** |
| Exercise • Create a java program which stores rollnumber, name and marks of a student in a text file. • It should be able to read and display these values along with the Grade of the student based on the following grading system 0-49 Fail 50-59 Pass 60- 69 Satisfactory 70-79 Good Above 80 Excellent |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **7** | **Week 7 :** | **Yes** |
| Run and provide output of following code  class A {  void m1(A a){  System.out.println("m1 method in class A");  } }  class B extends A {  public void m1(A a){  System.out.println("m1 method in class B");  } }  public class Test{  public static void main(String[] args){  A a = new A();  a.m1(a);  a.m1(new B());    B b = new B();  b.m1(null);    a = b;  a.m1(null);  a.m1(new A());  } } |
| **8** | **Week 8 :** | **Yes** |
| * Create a Graphical User Interface of your Project using the Swing Package * Add buttons menus and other swing items to make the interface interactive |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|
|
|
|
|
|
|
|
|
|
|

Note: All lab task details are provided in each lecture and lectures are accessible to students in the following link.

https://drive.google.com/drive/folders/1Weu6n6VSxRgc9FJRKbwJA5jMbN3KDtDg?usp=sharing