**Sagarmatha College of Science and Technology**

Lab sheets of

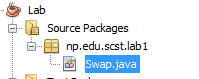
“Advanced Java Programming”

Course code: CSC-409

Prepared by: Er. Juned Alam

Lab conventions:

1. The Integrated Development Environment (IDE) to be used during the lab is Netbeans IDE 8.2, with Java SE Development Kit 8 (JDK 1.8) installed alongside it. (The later should be installed before the formal)
2. The project name for the entirety of the lab sessions must be named as “Lab”, containing within itself, the package in the format of : “np.edu.scst.lab*N*”, where *N* must be replaced by the corresponding lab number. Inside those packages, the corresponding lab classes must be present, named in such a way that they convey the message of the questions. For example, a class containing the program for odd-even should be named “OddEven.java”.



| **Steps to create this type of project structure:**   * *File>New Project>Java>Java Application>Next>Under Project Name, type "Lab" . Untick Create Main Class>Finish.* * *Once the project is created, right click on the source package and go to New>java package > under package name write "np.edu.scst.lab1" > Finish.* * *Now, right click on the newly created package> new> Java Class> and put your java class name there> Finish.* |
| --- |

1. If the use of multiple packages are required, those too should be created inside the above mentioned package.
2. We must follow the following naming conventions:

| Type | Naming convention |
| --- | --- |
| Project name, Class, Interface, Enum | **P**ascal**C**asing |
| Method name, variable | **c**amel**C**asing |
| Package name | **l**ower**c**asing |
| Constants | **CAPITALCASING** |

Please note: Students are required to supply their own values onto the variables unless stated.

Lab-1

**Basics of java programming Part -1**

1. Create a java class using notepad. Run the java file using the terminal.
2. **Write a Program** (**WAP**) which checks whether a number (initialized by you) is odd or even.
3. WAP which checks whether a number (initialized by you) is prime or not.
4. WAP to print 10 numbers using a while loop.
5. Create a one dimensional float array of size 10, fill it with values from 1 to 10 using a for loop. Now print the array using enhanced for loop (foreach loop).

Lab-2

**Basics of java programming Part -2**

1. WAP to sort the elements of an integer array in ascending order.
2. WAP using switch-case to tell the student’s division based on his/her marks.
3. WAP to display n terms of natural numbers and their sum.
4. WAP to print a multiplication table upto 10 of any number.
5. WAP to compute the area of a circle with radius r.

Lab-3

**Methods in java**

1. WAP containing method to find the smallest number among three numbers.
2. WAP containing a method to count all vowels in a string.
3. WAP containing method to compute the sum of the digits in an integer.
4. WAP containing method to calculate the area of a rectangle; the calculated area should be returned as a double data-type.
5. WAP containing a method to display the middle character of a String.

Lab-4

**Class and objects in java**

1. WAP that describes a class person. It should have instance variables to record name, age and salary. Create a person object. Set and display its instance variables.
2. WAP that creates a class circle with instance variables for the centre and the radius. Initialize and display its variables.
3. Modify experiment 1 to have a constructor in class circle to initialize its variables.
4. Modify experiment 2 to show constructor overloading.
5. WAP to display the use of this keyword.

Lab-5

**Inheritance, Polymorphism and abstraction in java**

1. WAP that implements method overloading.
2. WAP that shows passing an object as a parameter.
3. WAP that illustrates method overriding.
4. WAP to illustrate simple inheritance.
5. WAP to illustrate multilevel inheritance.

Lab-6

**Java class level keywords, Interfaces and access modifiers**

1. WAP illustrating all uses of super keywords.
2. WAP illustrating all uses of final keywords.
3. Create an abstract class shape. Let the rectangle and triangle inherit this shape class. Add necessary functions.
4. WAP to illustrate Interface Inheritance.
5. WAP to show the scope of the protected access modifier.

Lab-7

**Exception handling in java**

1. WAP to perform a division operation, with the handling of a possible exception that may arise.
2. WAP to print the list of characters array using for loop. Handle the possible exceptions that may arise.
3. WAP to check whether a person is eligible to vote or not. If the person is not of age, throw an exception with the message “You cannot vote.”
4. WAP to handle a checked exception with using throws keyword.
5. WAP to use try with finally block; put an exception there. Discuss the outcome.

Lab-8

**Threads and streams**

1. WAP to create 3 threads by implementing Runnable interface. Each thread should individually print a loop containing its name and priority.
2. Show a case of deadlock. Discuss on how it can be prevented?
3. WAP to read a file named “lab-read.txt” using a byte stream.
4. WAP to write the text “Hello world; this is written using character stream”, using character stream onto the file “lab-write.txt”.
5. WAP to create an object of a Mobile class. Now serialize this object and save it in “Serialize.ser” file.
6. WAP to deserialize the Mobile class from the previous question.

Lab-9

**JDBC and basic swing components**

1. WAP using JDBC-ODBC drivers to update a database table of your choice.
2. Write a java program using JDBC to extract the name of those avengers who live on planet "earth", assuming that the "avengers" table has four attributes (id, name, super\_power, planet).
3. Create any five components of java swing, now using GridLayout, arrange them in a formation of 2x3 grids inside a JPanel.

Lab-10

**Project-1 Part-1/3; Development of Desktop Application using swing and JDBC**

WAP to create a simple MVC (Model View Controller), CRUD (Create, read, update and delete) Student Management app swing application with the following description:

In a database, a student named table with columns (id, name, gender, address, faculty) should be created. Now using swing as a view, swing event handlers as controller and JDBC aided classes as model layer. The desktop app should contain labels, text fields, buttons, tables and other necessary components.

Lab-11

**Project-1 Part-2/3; Development of Desktop Application using swing and JDBC**

Continue with the previous lab question.

Lab-12

**Project-1 Part-3/3; Development of Desktop Application using swing and JDBC,**

**Java network programming**

1. Complete the lab exercise 10.
2. Send an email using javax.mail API.

Note: Depending on the lab progress one lab session may be added.

Lab-13

**Java network programming and JavaFX**

1. WAP for chatting between client and server using sockets.
2. WAP to create a simple yet functional JavaFX based desktop application containing various UI tools.

Lab-14

**Servlets and jsp basics**

1. WAP to create a servlet that displays "Hello World" when visiting a site.
2. WAP to create a jsp that displays "Hello World" when visiting a site.
3. WAP to create a simple web app that logs the number of times a user visited a site using cookies.
4. WAP to create a simple web app to pass the data from forms to the servlet using POST method.

Lab-15

**Project-2 Part-1/4; Simple Web Application Development**

Create a simple Student Management MVC Web Applications containing the following features:

1. Authentication for the teachers and students.
2. Authorization; such that the students have no access to page(s) in the web app containing some confidential information, for example a question collection page for teachers.
3. Both front-end and back-end data validation.
4. The use of bootstrap library for designing the view layer.
5. The use of sessions and/or cookies.
6. Of Course the CRUD functionality should be there.
7. There should be at least two many-to-many related tables.

Lab-16

**Project-2 Part-2/4; Simple Web Application Development**

Continue with the previous lab session.

Lab-17

**Project-2 Part-3/4; Simple Web Application Development**

Continue with the previous lab session.

Lab-18

**Project-2 Part-4/4; Simple Web Application Development**

Complete the previous lab session.

Note: Depending on the lab progress one or two lab sessions may be added.

.