**PROJECT BASED LAB REPORT**

**On**

**COURSE SELECTION FOR A +2 STUDENTS**

**Submitted in partial fulfilment of the**

**Requirements for the award of the Degree of**

**Bachelor of Technology**

**in**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**By**

**ROHINI PANDIRI 2100031934**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**K L University**

Green Fields, Vaddeswaram, Guntur District-522 502

**2016-2017**

**K L University**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**



***CERTIFICATE***

This is to certify that this project based lab report entitled **“COURSE SELECTION FOR A +2 STUDENTS”** is a bonafide work done by  **ROHINI PANDIRI 2100031934** in partial fulfilment of the requirements for the award of degree in **BACHELOR OF TECHNOLOGY** in **ELECTRONICS AND COMMUNICATION ENGINEERING**  during the Academic year 2016-2017.

**Faculty in Charge Head of the Department**

**Smt P.S.G.Aruna Sri A.C.S.C.SASTRY**

**Project guide**

**Dr.M.Jaya Bhaskar**

**K L University**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**



***DECLARATION***

We hereby declare that this project based lab report titled **“COURSE SELECTION FOR A +2 STUDENTS”** has been prepared by us in partial fulfilment of the requirements for the award of degree “**BACHELOR OF TECHNOLOGY in ELECTRONICS AND COMMUNICATION ENGNEERING**” during the Academic year 2016-2017.

We also declare that this project based lab report is of our own efforts and it has not been submitted to any other university for the award of any degree.

**ROHINI PANDIRI 2100031934**

**ACKNOWLEDGEMENT**

Our sincere thanks to **Dr.M.Jaya Bhaskar** in the Lab for their outstanding support through out the project for the successful completion of the work.

We express our gratitude to **A.C.S.C.SASTRY,** Head of the Department for Computer science and Engineering for providing us with adequate facilities, ways and means by which we are able to complete this project based Lab.

We would like to place on record the deep sense of gratitude to the honourable Vice Chancellor, K L University for providing the necessary facilities to carry the project based Lab.

Last, but not the least, we thank all Teaching and Non-Teaching Staff of our department and especially my classmates and my friends for their support in the completion of our project based Lab.

**ROHINI PANDIRI 2100031934**

**CONTENTS**

**s.no content**

**1. ABSTRACT**

**2. INTRODUCTION**

**3. FUNCTIONAL REQUIREMENTS**

**4. NON-FUNCTIONAL REQUIREMENTS**

**5. CODE**

**6. OUTPUT**

**Abstract**

**+2 course** selection is a software application to select the course after their +2. Now-a-days it is a crucial problem for students to select a course after +2. Firstly, we will take the student information like his name, parents’ names and their occupation. That will be the first module. In the second module we will collect the course he did in his +2 and what investment he can keep for his studies.

In the third module, we will give him/her the list of courses he can go for and if he is eligible for the scholarship or not. If he is eligible we will also calculate upto what amount he can get the scholarship.

All that will be in the third module. If the final module, we will print all the details of the course the student along with the course he/she is going to do**+**

**Keywords:**

* Registration
* Login
* Search for courses
* Available courses

**INTRODUCTION**

Java is the object oriented programming language. It is interpreted and high level language. In Java, program is written by creating class and object. Unlike C, in which preprocessors and pointers are used, Java does not make use of these components. Any programming language that uses object oriented concepts like Encapsulation, Polymorphism, Inheritance, Abstraction is called object oriented programming language and JAVA comes under the same category.

**Features of Java program**

C is the widely used language. It provides a lot of features that are given below.

1. **Simple**

Java language is simple because its syntax is similar to C++. Similarly, it has removed many confusing and rarely used features like explicit pointers, operator overloading, etc. it is not needed to remove unreferenced objects because there is Automatic Garbage Collection in Java.

1. **Object Oriented**

Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behavior. The basic concepts of OOPs are: Object, Class, Inheritance, Polymorphism, Abstraction and Encapsulation.

1. **Platform independent**

Java code can be run on multiple platforms e.g. Windows, Linux, Sun Solaris, Mac/OS etc. Java code is compiled by the compiler and converted into bytecode. This bytecode is a platform-independent code because it can be run on multiple platforms i.e. Write Once and Run Anywhere(WORA).

1. **Secured**

Java is secured because there is no use of explicit pointer and java programs run inside virtual machine sandbox.

1. **Robust**

Robust simply means strong. Java uses strong memory management. There is lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points make java robust.

1. **Architecture Neutral**

There are no implementation dependent features e.g. size of primitive types is fixed. Like- In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.

1. **Portable**

We may carry the java bytecode to any platform.

1. **High-performance**

Java is faster than traditional interpretation but still somewhat slower than a compiled language like C.

1. **Distributed**

We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

**10) Multi-threaded**

A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area.

**PROJECT DESCRIPTION**

The main purpose of the system is to automate course selection for a +2student system which maintains record ofcourses and their branches. It also provides easy method of searching the courses where one can search their course based on availability of their branches where it is located by just accessing the system.

Course selection for a +2 student system has been developed for automating the process of selecting and searching courses that are available in the system.

The system needs consistent flow of information about different courses available in the system within the course registration, any interruption in the flow of course registration can cause the final result to get pending. Following is the list of modules that comprise the system with the module description

**FUNCTIONAL REQUIREMENTS**

**Purpose**

The purpose of doing this project is to provide all the courses information for students who have completed their +2. The user can search easily anything regarding the course he want without moving here and there .This will make his time manageable and there will no waste of time in going out.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 |  | STUDENT information |  |
|  |  | Student details | Here, we will take the student information  like his name, parents’ names,address |
|  |  | Course taken in +2 | Here, the list of courses available in +2 is displayed |
|  |  | Courses after +2 | Here, the list of courses after +2 will be displayed |
|  |  | Sub Courses | Here sub courses will be displayed |
| 2 |  | Main Module |  |
|  |  | To display System | After all modules were integrated through files we display the complete information. |
|  |  | To process menu | Here **Processing Menu** can be done through if else statements with all the modules that are present in the file. |
|  |  | Initializations | Here **Initializations** can be done to methods, variables as well as objects which are present in classes |
|  |  | To access | To provide easy access we informed every thing to user which will able to access his details . we can be able to read the file and record of that file, and write that file where he can be able to do modifications for files that are present .. |

**NON-FUNCTIONAL REQUIREMENTS**

INTERFACE:

An interface is a reference type in Java. It is similar to class. It is a collection of abstract

methods. A class implements an interface, thereby inheriting the abstract methods of

the interface. Along with abstract methods, an interface may also contain constants, default

methods, static methods, and nested types.

INHERITANCE :

Inheritance in java is a mechanism in which one object acquires all the properties and

behaviors of parent object. The idea behind inheritance in java is that you can create new

classes that are built upon existing classes.

ABSTRACT CLASS:

A class that is declared with abstract keyword, is known as abstract class in java. It can have abstract and non-abstract methods (method with body).**Abstraction** is a process of hiding the implementation details and showing only functionality to the user. Another way, it shows only important things to the user and hides the internal details for example sending sms, you just type the text and send the message. You don't know the internal processing about the message delivery.

ABSTRACT METHOD:

An abstract method is a method that is declared, but contains no implementation. Abstract classes may not be instantiated, and require subclasses to provide implementations for the abstract methods. Let's look at an example of an **abstract** class, and an abstract method.

STATIC KEYWORD:

All instances share the same copy of the variable. A class variable can be accessed directly with the class, without the need to create a instance.

FINAL KEYWORD:

In the **Java** programming language, the **final keyword** is used in several different contexts to define an entity that can only be assigned once. Once a **final** variable has been assigned, it always contains the same value.

METHOD OVERRIDING:

In object oriented programming, is a language feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super classes or parent classes.

ARRAYS:

Java provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type

EXCEPTION HANDLING:

An exception (or exceptional event) is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore, these exceptions are to be handled.

**CODE:**

**import java.io.\*;**

**import java.util.Scanner;**

**class StudentDetails**

**{**

**String name,address,fathername,mothername;**

**String gender;**

**double percent,phonenumber;**

**void students()**

**{**

**System.out.println("enter student name");**

**Scanner sc=new Scanner(System.in);**

**name=sc.nextLine();**

**System.out.println("enter father name ");**

**fathername=sc.nextLine();**

**System.out.println("enter mother name");**

**mothername=sc.nextLine();**

**while(true)**

**{**

**System.out.println("enter gender");**

**gender=sc.nextLine();**

**if("m".equals(gender)||"M".equals(gender)||"f".equals(gender)||"F".equals(gender))**

**break;**

**System.out.println("enter correctly your gender");**

**}**

**System.out.println("enter your address");**

**address=sc.nextLine();**

**}**

**void selection()**

**{**

**int choice;**

**int op,num;**

**Scanner sc=new Scanner(System.in);**

**System.out.println("select your group in inter");**

**System.out.println("1.MPC");**

**System.out.println("2.BIPC");**

**System.out.println("3.MEC");**

**System.out.println("ENTER YOUR CHOICE");**

**while(true)**

**{**

**choice=sc.nextInt();**

**if(choice==1||choice==2||choice==3)**

**break;**

**else**

**System.out.println("enter choice correctly");**

**}**

**switch(choice)**

**{**

**case 1:**

**{**

**System.out.println("your list of courses");**

**System.out.println("1.B.TECH");**

**System.out.println("2.degree");**

**System.out.println("3.BBA");**

**System.out.println("enter your option");**

**while(true)**

**{**

**op=sc.nextInt();**

**if(op==1||op==2||op==3)**

**break;**

**else**

**System.out.println("enter option correctly");**

**}**

**switch(op)**

**{**

**case 1:**

**{**

**System.out.println("list of courses in b.tech are");**

**System.out.println("1.CSE");**

**System.out.println("2.ECE");**

**System.out.println("3.ECM");**

**System.out.println("4.MECHANICAL");**

**System.out.println("5.CIVIL");**

**System.out.println("enter your choice");**

**while(true)**

**{**

**num=sc.nextInt();**

**if(num==1||num==2||num==3||num==4||num==5)**

**break;**

**else**

**System.out.println("enter choice correctly");**

**}**

**switch(num)**

**{**

**case 1:**

**System.out.println(" selected course::CSE");**

**break;**

**case 2:**

**System.out.println("seleted course is ECE");**

**break;**

**case 3:**

**System.out.println("seleted course is ECM");**

**break;**

**case 4:**

**System.out.println("seleted course is mechanical");**

**break;**

**case 5:**

**System.out.println("seleted course is civil");**

**break;**

**}**

**break;**

**}**

**case 2:**

**{**

**System.out.println("list of courses in degree");**

**System.out.println("1.B.SC");**

**System.out.println("2.B.COM");**

**System.out.println("3.B.A");**

**System.out.println("enter your choice");**

**while(true)**

**{**

**num=sc.nextInt();**

**if(num==1||num==2||num==3)**

**break;**

**else**

**System.out.println("enter choice correctly");**

**}**

**switch(num)**

**{**

**case 1:**

**System.out.println("selected course is B.SC");**

**break;**

**case 2:**

**System.out.println("selected course iS B.COM");**

**break;**

**case 3:**

**System.out.println("selected course is B.A");**

**break;**

**}**

**break;**

**}**

**case 3:**

**System.out.println(" selected is BBA");**

**break;**

**default:**

**System.out.println("enter correct option");**

**break;**

**}**

**break;**

**}**

**case 2:**

**{**

**System.out.println("your list of courses");**

**System.out.println("1.MBBS");**

**System.out.println("2.BIOTECHNOLOGY");**

**System.out.println("3.PHARMACY");**

**System.out.println("ENTER YOUR OPTION");**

**while(true)**

**{**

**op=sc.nextInt();**

**if(op==1||op==2||op==3)**

**break;**

**else**

**System.out.println("enter option correctly");**

**}**

**switch(op)**

**{**

**case 1:**

**{**

**System.out.println("selected for mbbs");**

**break;**

**}**

**case 2:**

**{**

**System.out.println("your selected for biotechnology");**

**break;**

**}**

**case 3:**

**{**

**System.out.println("your selected for phramacy");**

**break;**

**}**

**default:**

**System.out.println("enter correct option");**

**break;**

**}**

**break;**

**}**

**case 3:**

**{**

**System.out.println("your selected couese");**

**System.out.println("1.CA");**

**System.out.println("2.BBA");**

**System.out.println("ENTER YOUR OPTION");**

**while(true)**

**{**

**op=sc.nextInt();**

**if(op==1||op==2)**

**break;**

**else**

**System.out.println("enter option correctly");**

**}**

**switch(op)**

**{**

**case 1:**

**{**

**System.out.println("your selected for CA");**

**break;**

**}**

**case 2:**

**{**

**System.out.println("your selected BBA");**

**break;**

**}**

**default:**

**System.out.println("enter correct option");**

**break;**

**}**

**break;**

**}**

**}**

**}**

**public void Students()**

**{**

**throw new UnsupportedOperationException("Not supported yet."); //Exception handiling**

**}**

**}**

**public class Project199**

**{**

**public static void main(String[] args)**

**{**

**StudentDetails sd=new StudentDetails();**

**sd.students();**

**try**

**{**

**PrintWriter pw = new PrintWriter("Myfile199.txt");**

**pw.print(sd.name+"\t");**

**pw.print(sd.fathername+"\t");**

**pw.print(sd.mothername+"\t");**

**pw.println(sd.gender);**

**pw.println(sd.address);**

**pw.close();**

**}**

**catch(FileNotFoundException e)**

**{**

**System.out.println(e.getMessage());**

**}**

**sd.selection();**

**try**

**{**

**Scanner s=new Scanner(new BufferedReader(new FileReader("Myfile199.txt")));**

**System.out.println( );**

**while(s.hasNext())**

**{**

**System.out.print(s.next()+"\t");**

**}**

**System.out.println( );**

**}**

**catch(FileNotFoundException e1)**

**{**

**System.out.print(e1.getMessage());**

**}**

**}**

**}**

**OUTPUT:**

**SCREEN SHOTS:**

