

## **INDEX**

<b>S.no.</b>	<b>Page</b>	<b>Topic</b>
1.	2	Team members
2.	2	Peer review
3.	3	Team Diary
4.	3	Overview
5.	4	Parts of programme
5(a)	4	Data base
5(b)	4	Main programme
(i)	4	Randomisation
(ii)	6	Reading the text file -
(iii)	6	File handling:
(iv)	9	Array
6.	9	Evaluation
7.	10	References
8.	11	Snapshots
9.	18	Time spent by Individual

-

### **TEAM MEMBERS:-**

<i>Name</i>	<i>roll no.</i>
<b>1.Subhadra dasgupta</b>	<b>- 115090024</b>
<b>2.Srijit chakrabarty</b>	<b>- 115280027</b>
<b>3.Smitesh behera</b>	<b>- 115060007</b>
<b>4.Surendar jaipal</b>	<b>- 110040027</b>
<b>5.Shubham sadavat</b>	<b>- 110040024</b>
<b>6.Sivalingam</b>	<b>- 11i060004</b>

## Peer Reveiw

Names	Mark
1.Subhadra dasgupta	9
2.Srijit chakrabarty	7
3.Shubham sadavat	7.5
4.Smitesh behera	6.5
5.Surendar jaipal	7
6.Sivalingam	3

## Team diary

Date	Duration & time	Topic	Members
2/10/2011	3.30-4.30 p.m	Deciding project	Shubham,Smitesh, Subhadra,Srijit, Sivalingam, Surendar.
15/10/2011	9.00-10.30 a.m	Writing project report Deciding modules, sketching algorithm.	all
22/10/2011	9.00-11.30 a.m	Final part of project report,work distribution	all
24/10/2011	3.5 hrs.	Coding main algorithm	Subhadra
27/10/2011	2.5 hrs	Coding main algorithm	Subhadra
29/10/2011	3.30-4.30 p.m	Discussion of progress and final implementation of algorithm.	Subhadra,Srijit Surendar,Smitiesh & our TA
30/10/2011	2 hrs.	Codig of display part	Srijit
31/10/2011	2 hrs.	Updating database	Shubham
3/11/2011	8.30-10.30 p.m	Final coding	whole group
4/11/2011	2 hrs.	Finalizing database	Surandar
6/11/2011	4.30-6.30 p.m	Final documentation of stage 2 project report.	Subhadra,Srijit Shubham,Surendar
<u>10/11/2011</u>	8.30-10.30	peer review and final modification.	whole group

## OVERVIEW:

This is a quiz on C++ programming language where :

All questions should be answered.

There are four choices to each question of which only one is correct .

You cannot skip any question

To move to next question you have to answer the current one.

To discontinue press q .

Time limit is 15 mins.

In this programme we have mainly use arrays for data file handling and random function has been used to select the questions randomly from the file.

## **PART OF THE PROGRAMMES:**

### **DATABASE**

We have a total data base containing 75 different questions and te corresponding choices and right answer.Text file have been written in data format such that first the question is written and after the question is finished the first choice start with opening “(. The 4 choices are written in four consecutive lines and in the next line the correct answer is stored.

### **QUESTION SELECTION PROCEDURE:-**

Our question selection procedure holds the following criteria:

Question must be such that not so simple of a person having good knowledge and not so much tough for a beginner.

Questions must be conceptual .

Questions must be such that one can understand better by attempting these questions.

Question must be such that so everybody can get better if one done the quiz in the stipulated time.

Questions must lies in the prescribed syllabus of 10+2 and first year b-tech.

Question must certainly enhances the knowledge after attempting these for everyone.

### **MAIN PROGRAMME:-**

There are mainly four parts -

**1)RANDOMISATION**

**2)READING THE TEXT FILE**

**3)DISPLAYING THE QUESTION AND TAKING THE ANSWERS**

**4)EVALUATION**

### **RANDOMISATION:-**

In this part we have written a programme whic will choose 20 numbers randomly out of 75.

These number corresponds to the question numbers in the text file which are to be displayed.

#### ABOUT THE FUNCTION AND THE HEADER FILES:-

"stdlib.h" is the header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others. It is compatible

with C++ and is known as cstdlib in C++. The name "stdlib" stands for "standard library".

We used rand,srand.

rand:It is used to generate a pseudo random numbers. A pseudorandom number generator (PRNG), also known as a deterministic random bit generator (DRBG),is an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's state, which includes a truly random seed. Although sequences that are closer to truly random can be generated using hardware random number generators, pseudorandom numbers are important in practice for their speed in number generation and their reproducibility, and they are thus central in applications such as simulations (e.g., of physical systems with the Monte Carlo method, in cryptography, and in procedural generation. Good statistical properties are a central requirement for the output of a PRNG, and common classes of suitable algorithms include linear congruential generators, lagged Fibonacci generators, and linear feedback shift registers.

In practice, the output from many common PRNGs exhibit artifacts which cause them to fail statistical pattern detection tests. These include:

1. Shorter than expected periods for some seed states (such seed states may be called 'weak' in this context);
2. Lack of uniformity of distribution for large amounts of generated numbers;
3. Correlation of successive values;
4. Poor dimensional distribution of the output sequence;
5. The distances between where certain values occur are distributed differently from those in a random sequence distribution.

"srand": It sets the pseudo random generator seed.

void srand ( unsigned int seed );

Initialize random number generator

The pseudo-random number generator is initialized using the argument passed as seed.

For every different seed value used in a call to srand, the pseudo-random number generator can be expected to generate a different succession of results in the subsequent calls to rand.

Two different initializations with the same seed, instructs the pseudo-random generator to generate the same succession of results for the subsequent calls to rand in both cases.

If seed is set to 1, the generator is reinitialized to its initial value and produces the same values as before any call to rand or srand.

In order to generate random-like numbers, srand is usually initialized to some distinctive value, like those related with the execution time. For example, the value returned by the function time (declared in header <ctime>) is different each second, which is distinctive enough for most randoming needs.

Parameters

seed

An integer value to be used as seed by the pseudo-random number generator algorithm.

Return Value

(none)

and we have also used the time function to initialize the counter for random numbers

Standard library and about the function

<ctime>

C date and time operations refer to a group of functions in the standard library of the C programming language implementing date and time manipulation operations.

Functions used:time()

time: The time() function shall return the value of time.

## READING THE TEXT FILE-

We have written the programme which will read the text file line by line using “fgets” function. It will first start reading the file such that it stores the data in two dimensional array “quest” till it encounters ‘\n’ in first line. As soon as it encounters “\n” as a first character of a line it starts storing the text in another two dimensional array “ans”. Since we know that the answers are in the four consecutive lines using a for loop in four iteration we put the whole set of choices in array “ans”. Then it takes the answer in “op” array and again it starts repeating the whole process to take the next set of problems choices and answers in the next iteration. Since we know that there are 75 questions so we use a for loop which will perform 75 iterations

In this part we used the 'fopen' function to open the file and then “ fgets” to read the file line by line and after this process is over we use “ fclose” to close the file. We now discuss what are files ,what is the header file and what are the modes in which it is opened.

## FILE HANDLING:

Files are very important for storing information permanently. We store information in files for many purposes, like data processing by our programs

### about files-

Abstractly, a file is a collection of bytes stored on a secondary storage device, which is generally a disk of some kind. The collection of bytes may be interpreted, for example, as characters, words, lines, paragraphs and pages from a textual document; fields and records belonging to a database; or pixels from a graphical image. A file is simply a machine decipherable storage media where programs and data are stored for machine usage.

Essentially there are two kinds of files that programmers deal with,

#### **1. Text files.**

#### **2. Binary files.**

But in this project we are only dealing with the text files.

### **Text files**

A text file can be a stream of characters that a computer can process sequentially. It is not only processed sequentially but only in forward direction. For this reason a text file is usually opened for only one kind of operation (reading, writing, or appending) at any given time.

Similarly, since text files only process characters, they can only read or write data one character at a time. (In C Programming Language, Functions are provided that deal with lines of text, but these still essentially process data one character at a time.)

In this project we have created a text file and also used the commands to input into file and to get output from the file, here below we have defined how to creat a file and output some data,

The type FILE is used for a file variable and is defined in the stdio.h file. It is used to define a file pointer for use in file operations. Before we can write to a file, we must open it. What this really means is that we must tell the system that we want to write to a file and what the file name is. We do this with the fopen() function illustrated in the first line of the program. The file pointer, fp in our case, points to the file and two arguments are required in the parentheses, the file name first, followed by the file type.

The file name is any valid Terminal file name, and can be expressed in upper or lower case letters, or even mixed if you so desire. It is enclosed in double quotes. For this example we have chosen the name All\_questions.txt. This file should not exist on your disk at this time. If you have a file with this name, you should change its name or move it because when we execute this program, its contents will be erased. If you don't have a file by this name, that is good because we will create one and put some data into it. You are permitted to include a directory with the file name. The directory must, of course, be a valid directory otherwise an error will occur. Also, because of the way C++ handles literal strings, the directory separation character must be written twice. For example, if the file is to be stored in the \PROJECTS sub directory then the file name should be entered as "\\PROJECTS\\All\_question.txt". The second parameter is the file attribute and can be any of three letters, r, w, or a, and must be lower case.

## **MODES IN WHICH FILES CAN BE OPENED**

### **Reading (r)**

When an r is used, the file is opened for reading. Using the r indicates that the file is assumed to be a text file. Opening a file for reading requires that the file already exist. If it does not exist, the file pointer will be set to NULL and can be checked by the program.

### **Writing (w)**

When a file is opened for writing, it will be created if it does not already exist and it will be reset if it does, resulting in the deletion of any data already there. Using the w indicates that the file is assumed to be a text file.

### **Appending (a)**

When a file is opened for appending, it will be created if it does not already exist and it will be initially empty. If it does exist, the data input point will be positioned at the end of the present data so that any new data will be added to any data that already exists in the file. Using the a indicates that the file is assumed to be a text file.

Although in our program we need to use only reading of files.

When an r is used, the file is opened for reading. Using the r indicates that the file is assumed to be a text file. Opening a file for reading requires that the file already exist. If it does not exist, the file pointer will be set to NULL and can be checked by the program.

When a file is opened for writing, it will be created if it does not already exist and it will be reset if it does, resulting in the deletion of any data already there. Using the w indicates that the file is assumed to be a text file.

When a file is opened for appending, it will be created if it does not already exist and it will be

initially empty. If it does exist, the data input point will be positioned at the end of the present data so that any new data will be added to any data that already exists in the file. Using the `a` indicates that the file is assumed to be a text file.

## **Outputting to the file:-**

The job of actually outputting to the file is nearly identical to the outputting we have already done to the standard output device. The only real differences are the new function names and the addition of the file pointer as one of the function arguments. In the example program, `fprintf` replaces our familiar `printf` function name, and the file pointer defined earlier is the first argument within the parentheses. The remainder of the statement looks like, and in fact is identical to, the `printf` statement.

## **Closing a file:-**

To close a file we simply use the function `fclose` with the file pointer in the parentheses. We can open a file for writing, close it, and reopen it for reading, then close it, and open it again for appending, etc. Each time we open it, you could use the same file pointer, or you could use a different one. The file pointer is simply a tool that we use to point to a file and we decide what file it will point to.

## **READING A FILE FROM TEXT:-**

Now for our first program that reads from a file. This program begins with the familiar include, some data definitions, and the file opening statement which should require no explanation except for the fact that an `r` is used here because we want to read it.

Use of file handling in our project

In our project we have used file handling to read a text file containing questions with 4 options followed by the correct answers.

Firstly we read the whole text file and save it in the array named as `data`. Then we read only a single question from `data` array and store that question in an another array `quest`. After the question is stored in array, the 4 choices are read and store in the array `ans`.

Every question in the text file followed by the correct answer. So after reading the choices the correct answer is read and store in another array `op`. After displaying every question user is asked to input his answer. The answer given by the user is also stored in an another array. Every time a single question is displayed on the screen. At last, after reading the input by user the file is closed. When the test is over, the two arrays, one which contain the correct solution and the other which contain the answer given by user are compared so as to know how many questions are correctly answered and how many are wrongly answered by user.

## **DISPLAYING THE QUESTION AND TAKING THE ANSWERS -**

This is done by using another “for” loop which will display the questions and the choices. Such that if the randomly selected number is “`i`” then it will display the `i`’th question and the corresponding choices. Take the answer from the user compare it with the correct answer with “`op`” array and if the answer is right it will increase the counter for right by one and if it is wrong then it will increase the counter by one. And if at any stage answer received is “`q`” then this part of programme will stop and move on to the evaluation part.



## **ARRAY-**

Arrays are used to implement mathematical vectors and matrices, as well as other kinds of rectangular tables. Many databases, small and large, consist of (or include) one-dimensional arrays whose elements are records.

Arrays are used to implement other data structures, such as heaps, hash tables, deques, queues, stacks and strings.

## **ONE DIMENSIONAL ARRAY**

A one-dimensional array (or single dimension array) is a type of linear array. Accessing its elements involves a single subscript which can either represent a row or column index.

## **TWO DIMENSIONAL ARRAY**

For a two-dimensional array, the element with indices  $i, j$  would have address  $B + c \cdot i + d \cdot j$ , where the coefficients  $c$  and  $d$  are the row and column address increments, respectively.

## **MULTIDIMENSIONAL ARRAY**

Multidimensional arrays can be described as "arrays of arrays". For example, a bidimensional array can be imagined as a bidimensional table made of elements, all of them of a same uniform data type.

<cctype>

C character classification is an operation provided by a group of functions in the C++ Standard Library for the C++ programming language. These functions are used to test characters for membership in a particular class of characters, such as alphabetic characters, control characters, etc. Both single-byte, and wide characters are supported.

Functions used: toupper.

toupper: toupper() converts the letter  $c$  to upper case, if possible.

tolower() converts the letter  $c$  to lower case, if possible.

If  $c$  is not an unsigned char value, or EOF, the behavior of these functions is undefined.

## **EVALUATION:-**

When the user have provided answers for all question or the time limit for quiz is over, then the the answees provided by user are compared with correct answers and finally the total marks gained by the user are showed to him.

The criteria for marking is that for every wrong answer user is awarded with -1

and for every correct answer user is awarded +4 marks. So total score of user is given as  $-(\text{total number of correct answer given by user}) + 4 \times (\text{number of question answered wrong by the user})$ .

## **REFERENCES :**

### **Books:**

A.K.Sharma,Dhanpat rai publications

Sumita Arora,Dhanpat rai publications

### **Websites:**

[www.cprogramming.com](http://www.cprogramming.com)

[www.cquestionbank.blogspot.com](http://www.cquestionbank.blogspot.com)

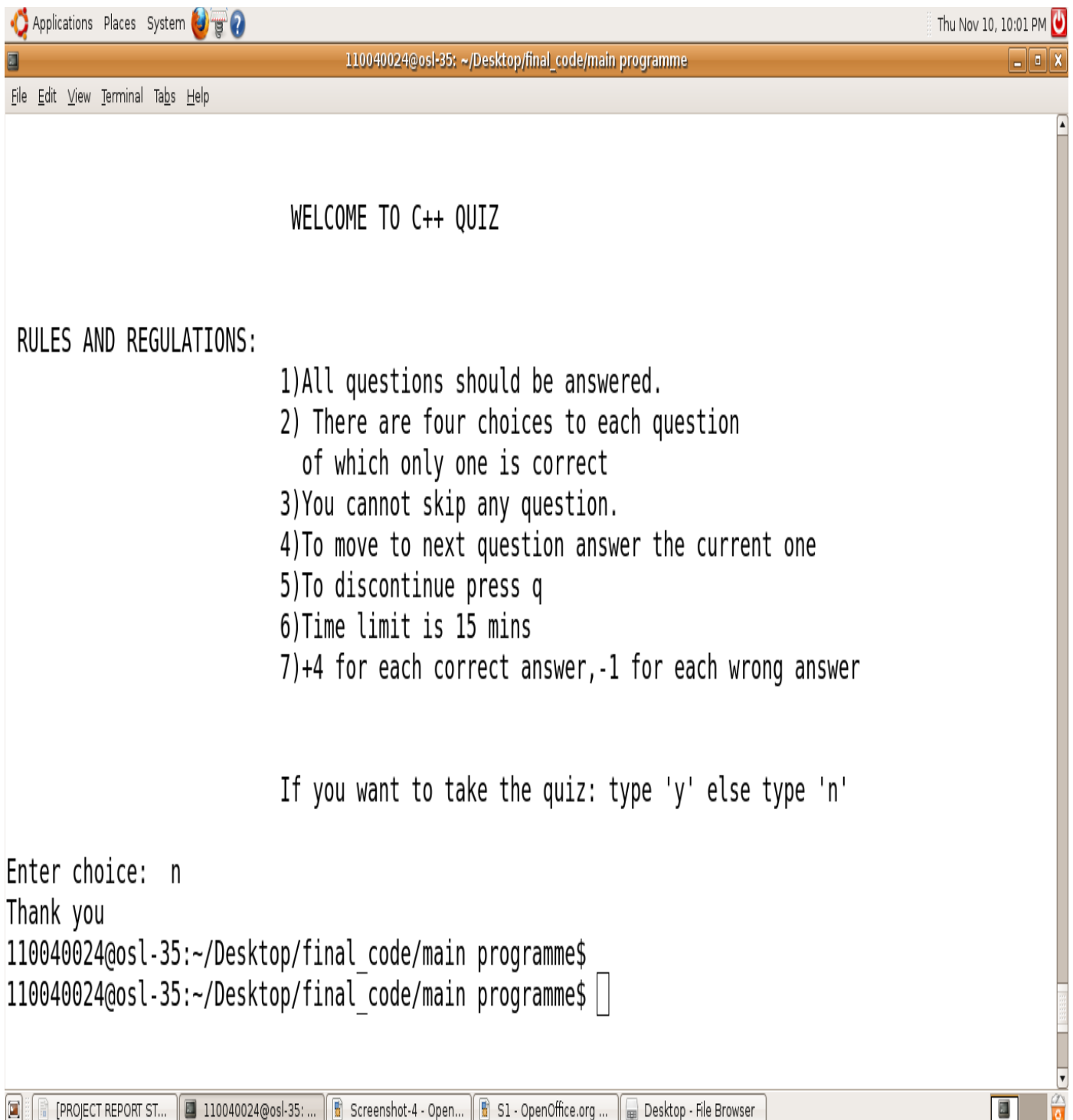
[www.genesisinsoft.com/QBCPP](http://www.genesisinsoft.com/QBCPP).

[www.mycppquiz.com](http://www.mycppquiz.com)

[www.wikipedia.com](http://www.wikipedia.com)

# **SNAPSHOTS:-**

1. when user do not want to take the test:



```
Applications Places System ? Thu Nov 10, 10:01 PM
110040024@osl-35: ~/Desktop/final_code/main programme
File Edit View Terminal Tabs Help

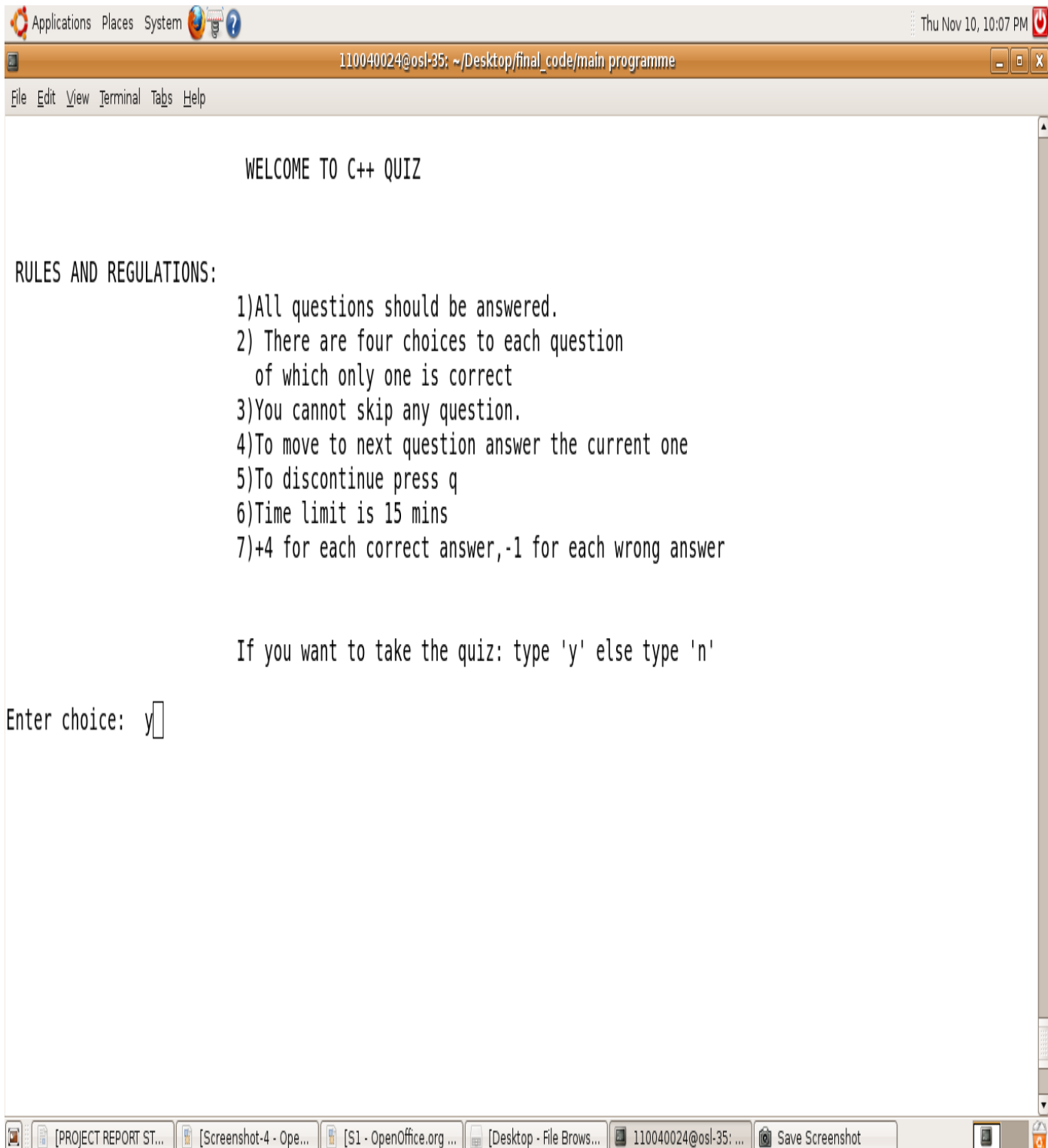
WELCOME TO C++ QUIZ

RULES AND REGULATIONS:
1)All questions should be answered.
2) There are four choices to each question
   of which only one is correct
3)You cannot skip any question.
4)To move to next question answer the current one
5)To discontinue press q
6)Time limit is 15 mins
7)+4 for each correct answer,-1 for each wrong answer

If you want to take the quiz: type 'y' else type 'n'

Enter choice: n
Thank you
110040024@osl-35:~/Desktop/final_code/main programme$
110040024@osl-35:~/Desktop/final_code/main programme$
```

## 2. When user wants to quit at intermediate stage



```
Applications Places System ?
110040024@osl-35: ~/Desktop/final_code/main programme
Thu Nov 10, 10:07 PM

File Edit View Terminal Tabs Help

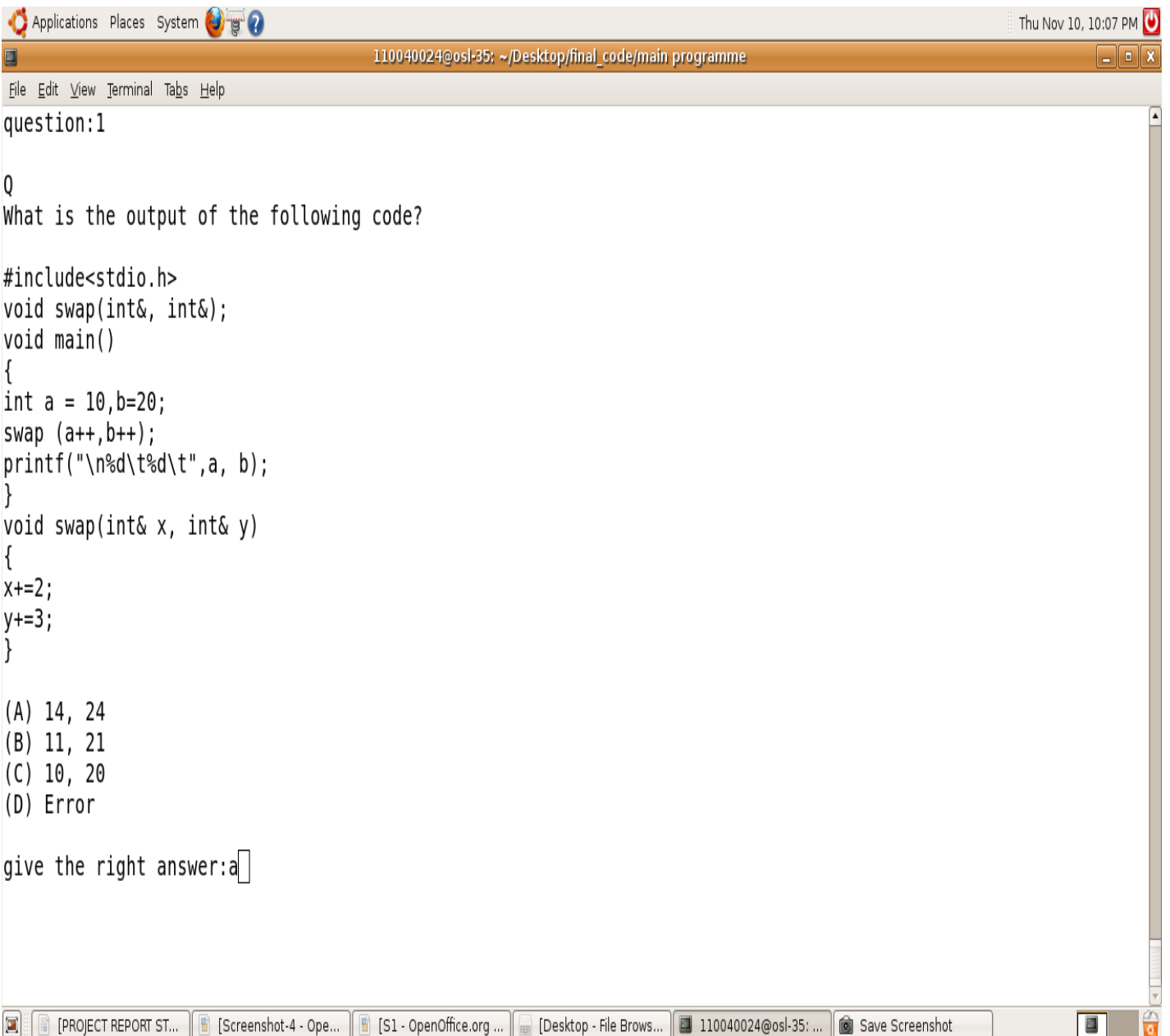
WELCOME TO C++ QUIZ

RULES AND REGULATIONS:
1)All questions should be answered.
2) There are four choices to each question
  of which only one is correct
3)You cannot skip any question.
4)To move to next question answer the current one
5)To discontinue press q
6)Time limit is 15 mins
7)+4 for each correct answer,-1 for each wrong answer

If you want to take the quiz: type 'y' else type 'n'

Enter choice: y
```

[PROJECT REPORT ST... [Screenshot-4 - Ope... [S1 - OpenOffice.org ... [Desktop - File Brows... 110040024@osl-35: ... Save Screenshot



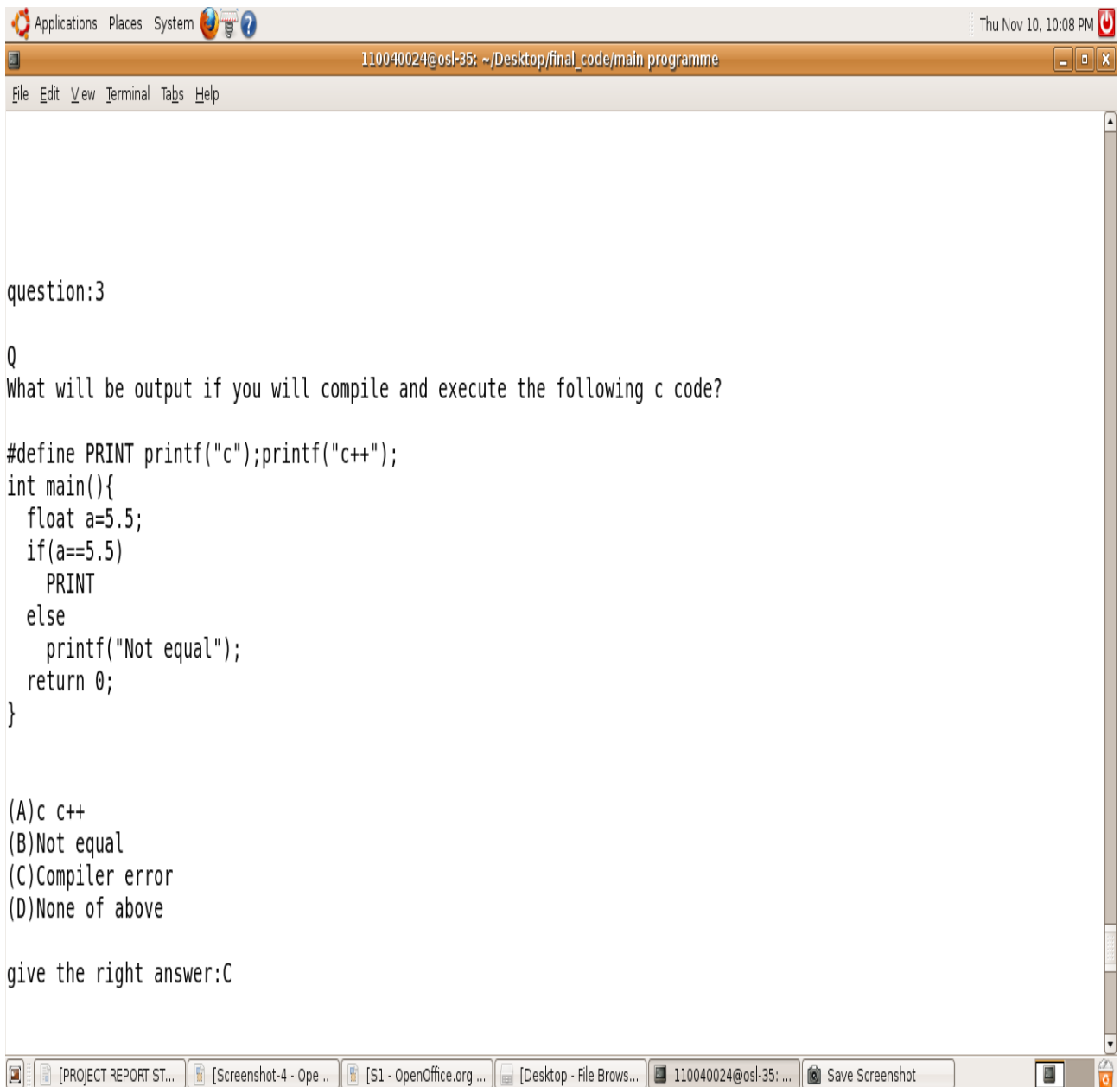
question:2

Q

What is the size of empty class?

- (A) 0 bytes
- (B) 2 bytes
- (C) 1 byte
- (D) 4 bytes

give the right answer:b



question:4

```
Q
#include<stdio.h>
int main(){
    char *str;
    scanf("%[^\n]",str);
    printf("%s",str);
    return 0;
}
```

- (A)It will accept a word as a string from user.
- (B)It will accept a sentence as a string from user.
- (C)It will accept a paragraph as a string from user.
- (D)Compiler error

give the right answer:q



110040024@osl-35: ~/Desktop/final\_code/main programme

File Edit View Terminal Tabs Help

You answered 1 questions correctly and 2 questions wrongly

\*\*\*\*\*

YOU SCORED :2

\*\*\*\*\*

110040024@osl-35:~/Desktop/final\_code/main programme\$

## TIME SPENT BY INDIVIDUAL

PERSON	DISCUSSION	DESING	PROGRAMMING	TESTING	DOC	MISC.
1.Subhadra	6	4	14	2	4	-
2.Srijit	6	2	5	2	4	-
3.Smitesh	6	2.30	2.30	2	5	-
4.Surendar	6	2.30	1.30	2	7	-
5.Shubham	5	2	2.30	2	8:30	-
6.Sivalingam	4	1	1	2	1.30	-