**CODE:**

#include <iostream>

#include <conio.h>

#include <stdlib.h> //for srand() function

#include <fstream> //for file handling

#include <time.h> //for time used in srand() function

#include<string.h> //for string

#include<windows.h>

//Maximum count of questions present in the file

int max\_ques = 92;

int cnt=1;

//Global Variables to store the count of the papers the user wants to generate and questions in each of them

long int enroll;

int paper\_count = 0;

using namespace std;

class question\_papers

{

public:

//function to show the randomly generated question from file

void generator(int m);

};

void question\_papers::generator(int m) //input m comes from the random function below

{

//string variable named line created to store each individual question generated, fin object is created of fstream class

string line;

ifstream fin;

ofstream out;

//object f fstream class used to read from an existing file names questions

fin.open("Questions.txt",ios::in);

out.open("output.txt",ios::app);

int con=1;

if (!fin)

{

cout << "Not able to display the file"; //if file not found

}

else

{

for (int i = 1; i <= max\_ques; i++)

{

getline(fin, line);

//If count matches the line number we want to display, only then print the line

if (i == m)

{

cout << "\t\t\t\t\t"<< "Q"<<cnt<< line << "\n";

out<<"\t\t\t\t\t"<< "Q"<<cnt<<line<<endl;

cnt++;

}

}

}

out.close();

}

//to display a basic line

void line()

{

cout << "\n\t\t\t\t\t";

for(int i=0;i<100;i++)

{

cout<< "-";

Sleep(20);

}

cout<<endl;

}

//display the details of papers

void format()

{

ofstream out1;

out1.open("output.txt",ios::app);

cout<<"\t\t\t\t\t\t\t\t\t\t\tGOVERNMENT POLYTECNIC PUNE 411061\n";

out1<<"\n\n\n\t\t\t\t\t\t\t\t\t\t\tGOVERNMENT POLYTECNIC PUNE 411061\n";

cout<<"\t\t\t\t\t\t\t\t\t\t(An Autonomous Institute of Govt. of Maharashtra)\n";

out1<<"\t\t\t\t\t\t\t\t\t\t(An Autonomous Institute of Govt. of Maharashtra)\n";

cout<<"\t\t\t\t\t\t\t\t\t\t\t\tTERM EXAM ODD2019\n";

out1<<"\t\t\t\t\t\t\t\t\t\t\t\tTERM EXAM ODD2019\n";

cout<<"\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tEnroll. No:"<<enroll;

out1<<"\t\t\t\t\t\t\t\t\t\t\t\t\t\t\tEnroll. No:"<<enroll;

cout << "\n\t\t\t\t\t";

out1 << "\n\t\t\t\t\t";

for(int i=0;i<100;i++)

{

cout<< "-";

out1<<"-";

Sleep(20);

}

cout<<"\n";

out1<<"\n";

cout<<"\t\t\t\t\tProgramme: Diploma in Information Technology\n"<<endl;

out1<<"\n\t\t\t\t\tProgramme: Diploma in Information Technology\n"<<endl;

cout<<"\t\t\t\t\tCourse Name: Object Oriented Programming: C++\t\t\t\t\tTime Allowed: 3 Hrs.\n";

out1<<"\n\t\t\t\t\tCourse Name: Object Oriented Programming: C++\t\t\t\t\tTime Allowed: 3 Hrs.\n";

cout<<"\t\t\t\t\tCourse Code:CM388\t\t\t\t\t\t\t\tMax Marks: 80\n";

out1<<"\n\t\t\t\t\tCourse Code:CM388\t\t\t\t\t\t\t\tMax Marks: 80\n";

cout << "\n\t\t\t\t\t";

out1 << "\n\t\t\t\t\t";

for(int i=0;i<100;i++)

{

cout<< "-";

out1<<"-";

Sleep(20);

}

cout<<"\n\n\n";

out1<<"\n\n\n";

}

//to choose the heading for each main questions

void question(int x)

{

ofstream out1;

out1.open("output.txt",ios::app);

switch(x)

{

case 1: cout<<"\n\n\t\t\t\t\tQ.1 Attempt any FOUR of the following.\t\t\t\t\t\t\t\t\t16M\n\n";

out1<< "\n\n\t\t\t\t\tQ.1 Attempt any FOUR of the following.\t\t\t\t\t\t\t\t\t16M\n\n";

break;

case 2: cout<<"\n\n\t\t\t\t\tQ.2 Attempt any TWO of the following. \t\t\t\t\t\t\t\t\t12M\n\n";

out1<< "\n\n\t\t\t\t\tQ.1 Attempt any FOUR of the following.\t\t\t\t\t\t\t\t\t16M\n\n";

break;

case 3: cout<<"\n\n\t\t\t\t\tQ.3 Attempt any TWO of the following. \t\t\t\t\t\t\t\t\t12M\n\n";

out1<< "\n\n\t\t\t\t\tQ.3 Attempt any TWO of the following. \t\t\t\t\t\t\t\t\t12M\n\n";

break;

case 4: cout<<"\n\n\t\t\t\t\tQ.4 Attempt any FOUR of the following.\t\t\t\t\t\t\t\t\t16M\n\n";

out1<< "\n\n\t\t\t\t\tQ.4 Attempt any FOUR of the following.\t\t\t\t\t\t\t\t\t16M\n\n";

break;

case 5: cout<<"\n\n\t\t\t\t\tQ.5 Attempt any TWO of the following. \t\t\t\t\t\t\t\t\t12M\n\n";

out1<< "\n\n\t\t\t\t\tQ.5 Attempt any TWO of the following. \t\t\t\t\t\t\t\t\t12M\n\n";

break;

case 6: cout<<"\n\n\t\t\t\t\tQ.6 Attempt any THREE of the following\t\t\t\t\t\t\t\t\t12M\n\n";

out1<< "\n\n\t\t\t\t\tQ.6 Attempt any THREE of the following\t\t\t\t\t\t\t\t\t12M\n\n";

break;

}

}

//Get user input for No. of Papers and Questions in each paper

void get\_no()

{

cout << "\n\t# Enter the number of papers to be generated: ";

cin >> paper\_count;

cout << "\n\t# Enter enrollment number: ";

cin >> enroll;

}

//Function to randomly generate question numbers

int random()

{

srand(time(0));

return ((rand() % max\_ques));

}

int main()

{

//Array of objects : Each object instance represents a paper

question\_papers Q[paper\_count];

line();

cout << "\t\t\t\t\t\t\t\t\t\tWelcome to Random Question Paper Generator";

line();

get\_no();

//array to store the numbers already generated by rand() function to avoid repetition

int no[100] = {0};

// to control the array input

static int b;

int m;

// clears screen

system("cls");

//loop for no of papers

for (int i = 0; i < paper\_count; i++)

{

//b initialized to zero before generation of each paper

b=0;

cout << "=================================================================================================== PAPER " << i + 1 << " ===================================================================================================\n\n\n\n\n";

format();

// loop for main questions

for(int q=1;q<=6;q++)

{

question(q);

cnt=1;

// loop for subquestions within main questions

for (int j =1; j <=4 ; j++)

{

//Generate Question numbers

repeat: // label

m = random();

//If a question number is repeated, regenerate it so that a question doesn't appear more than once in a single question paper

for (int k = 0; k < max\_ques; k++)

{

if (no[k] == m)

{

goto repeat;

}

}

//Save the question number to array no[]

no[b] = m;

b++;

//Send the generated number to generator() function for reading the appropriate question from file

Q[i].generator(m);

}

}

ofstream out1;

out1.open("output.txt",ios::app);

cout<<"\n\n\n\n\n\n\n\n\n";

out1<<"\n\n\n\n\n\n\n\n\n";

}

//Hold the current screen until the user presses a key

cout<<"\t\t\t\t\t";system("pause");

return 0;

}