**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management and Engineering, Mumbai**

**Department of Electronics & Telecommunication**



**Programming for Problem Solving (Exp 9 - 1)**

|  |  |
| --- | --- |
| **Roll No: J001** | **Name: Adith Ramakrishna** |
| **Program: B. Tech Data Science (1st)** | **Batch: J1** |
| **Date of Experiment: 12/11/2022** | **Date of Submission: 19/11/2022** |

**Task 1:**

**Output:**

AbdulKalam 1931

**Task 2:**

**Code:**

#include <iostream>

using namespace std;

struct student {

int code = 0;

char name[30] = "";

float marks = 0;

int dept\_number = 0;

}

students;

int main() {

cout << "Enter details for new student\n\nCode: ";

cin >> students.code;

cout << "Name: ";

cin >> students.name;

cout << "Marks: ";

cin >> students.marks;

cout << "Department No: ";

cin >> students.dept\_number;

cout << endl;

cout << "\n\nStudent Details:\n\nName: " << students.name << "\nCode: " << students.code << "\nMarks: " << students.marks;

if (students.dept\_number == 1) {

cout << "\nDepartment: IT";

} else if (students.dept\_number == 2) {

cout << "\nDepartment: COMP";

} else if (students.dept\_number == 3) {

cout << "\nDepartment: EXTC";

} else if (students.dept\_number == 4) {

cout << "\nDepartment: Data Science";

}

}

**Task 3:**

#include <iostream>

#include <cstring>

using namespace std;

struct employee {

char name[30] = "";

int id = 0;

int experience = 0;

long long salary = 0;

}

emp[100];

int main() {

strcpy(emp[0].name, "Adith");

emp[0].id = 1;

emp[0].experience = 5;

emp[0].salary = 10000;

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

if (emp[i].salary < 100000 && emp[i].experience >= 5) {

cout << "\n\nEmployee Details:\n\nName: " << emp[i].name << "\nID: " << emp[i].id << "\nExperience: " << emp[i].experience << "\nSalary: " << emp[i].salary;

}

}

}

**Task 4:**

**Output:**

The size of staff structure variable 48

**Task 5:**

**Code:**

#include <iostream>

#include <algorithm>

using namespace std;

struct student {

char name[30] = "";

int roll\_no = 0;

float marks\_1 = 0;

float marks\_2 = 0;

float marks\_3 = 0;

float grand\_total = 0;

};

bool compareMarks(student g1, student g2) {

if (g1.grand\_total < g2.grand\_total) {

return true;

}

return false;

}

int main() {

int n;

cout << "Enter number of students: ";

cin >> n;

cout << "\n\n";

student students[n];

for (int i = 0; i < n; i++) {

cout << "Enter details for student " << i + 1 << "\nName: ";

cin >> students[i].name;

cout << "Roll No: ";

cin >> students[i].roll\_no;

cout << "Marks 1: ";

cin >> students[i].marks\_1;

cout << "Marks 2: ";

cin >> students[i].marks\_2;

cout << "Marks 3: ";

cin >> students[i].marks\_3;

students[i].grand\_total = students[i].marks\_1 + students[i].marks\_2 + students[i].marks\_3;

cout << endl;

}

sort(students, students + n, compareMarks);

}<< endl;

}

}

**Homework Questions:**

**1:**

**Code:**

#include <iostream>

using namespace std;

struct computer {

char cpu\_type[30] = "";

long long hard\_disk\_size = 0;

char keyboard\_type[30] = "";

char mouse\_type[30] = "";

char monitor\_type[30] = "";

}

office\_comp[50];

int main() {

for (int i = 0; i < 50; i++) {

cout << "Enter details for Computer " << i + 1 << "\nCPU Type: ";

cin >> office\_comp[i].cpu\_type;

cout << "Hard Disk Size (GB): ";

cin >> office\_comp[i].hard\_disk\_size;

cout << "Keyboard Type: ";

cin >> office\_comp[i].keyboard\_type;

cout << "Mouse Type: ";

cin >> office\_comp[i].mouse\_type;

cout << "Monitor Type: ";

cin >> office\_comp[i].monitor\_type;

cout << "\n\n";

}

for (int i = 0; i < 50; i++) {

if (office\_comp[i].hard\_disk\_size > 8) {

cout << "\n\nComputer Details:\n\nCPU Type: " << office\_comp[i].cpu\_type << "\nHard Disk Size (GB): " << office\_comp[i].hard\_disk\_size << "\nKeyboard Type: " << office\_comp[i].keyboard\_type << "\nMouse Type: " << office\_comp[i].mouse\_type << "\nMonitor Type: " << office\_comp[i].monitor\_type;

}

}

}

**2:**

**Structures are a way to group several related variables of any data type combinations into one place. Each variable in the structure is known as a member of the structure.**

**3:**

**A structure inside another structure is called nested structure**

**4:**

**Code:**

#include <iostream>

using namespace std;

struct student {

int code = 0;

char name[30] = "";

float marks = 0;

int dept\_number = 0;

};

int main() {

int n;

cout << "Enter number of students: ";

cin >> n;

cout << "\n\n";

student students[n];

for (int i = 0; i < n; i++) {

cout << "Enter details for student " << i + 1 << "\n\nCode: ";

cin >> students[i].code;

cout << "Name: ";

cin >> students[i].name;

cout << "Marks: ";

cin >> students[i].marks;

cout << "Department No: ";

cin >> students[i].dept\_number;

cout << "\n\n";

}

char input, repeat = 'y';

int i, max = 0, min = 0, dept\_num = 0;

do {

cout << "\n\nStudent Management System:\n\na. Display Highest and Lowest Marks\nb. Display Department Wise Student Info\n\n";

cin >> input;

switch (input) {

case 'a':

max = students[0].marks;

min = students[0].marks;

for (i = 1; i < n; i++) {

if (students[i].marks > max) {

max = i;

}

if (students[i].marks < min) {

min = i;

}

}

cout << "\nHighest Marks:\nName: " << students[max].name << "\nMarks: " << students[max].marks << endl;

cout << "\nLowest Marks:\nName: " << students[min].name << "\nMarks: " << students[min].marks << endl;

break;

case 'b':

cout << "Enter Department Number: ";

cin >> dept\_num;

cout << "\n\n";

for (i = 0; i < n; i++) {

if (students[i].dept\_number == dept\_num) {

cout << "\n\nStudent Details:\n\nName: " << students[i].name << "\nCode: " << students[i].code << "\nMarks: " << students[i].marks;

if (students[i].dept\_number == 1) {

cout << "\nDepartment: IT";

} else if (students[i].dept\_number == 2) {

cout << "\nDepartment: COMP";

} else if (students[i].dept\_number == 3) {

cout << "\nDepartment: EXTC";

} else if (students[i].dept\_number == 4) {

cout << "\nDepartment: Data Science";

}

}

}

break;

default:

cout << "Invalid Option!";

break;

}

cout << "\n\n\nWould you like to continue? (Y/n)\n";

cin >> repeat;

} while (repeat != 'n' && repeat != 'N');

}