Department of Computer Science

Maqsood Ahmed 38186

PF Assignment : 03

Question 01[CLO3: C3][20]

Part A

Write a program of currency converter using functions:
1. Press 1 :Call function which convert Pakistani Rs in to euro
2. Press 2 :Call function which convert euro to yen
3. Press 3 :Dollar in to euro
4. Press 4 :Pakistani Rs to dollar
5. Press 5 :Dollar in to yen

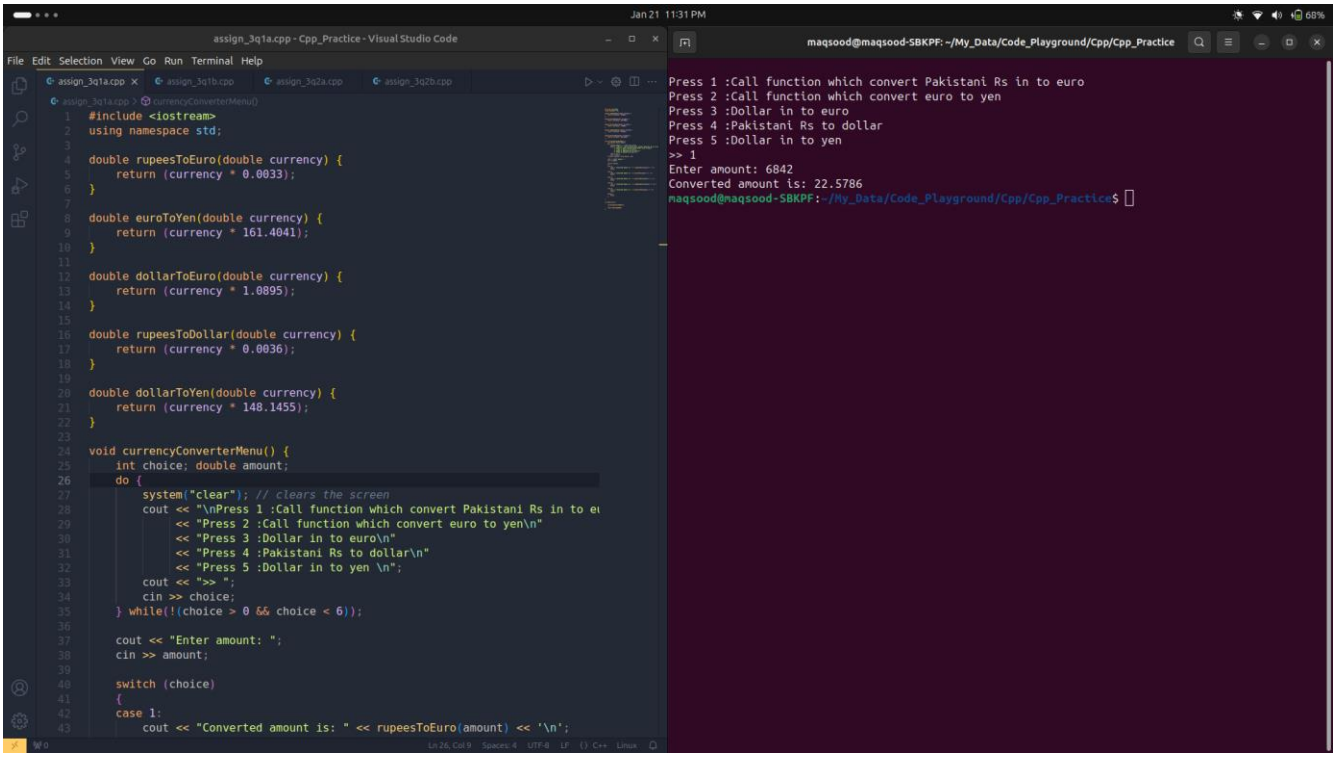
Source Code:

```
#include <iostream>
using namespace std;

double rupeesToEuro(double currency) {
   return (currency * 0.0033);
}

double euroToYen(double currency) {
   return (currency * 161.4041);
}

double dollarToEuro(double currency) {
   return (currency * 1.0895);
}

double rupeesToDollar(double currency) {
   return (currency * 0.0036);
}

double dollarToYen(double currency) {
   return (currency * 148.1455);
}

void currencyConverterMenu() {
   int choice; double amount;
   do {
```

```
        system("clear"); // clears the screen
        cout << "\nPress 1 :Call function which convert Pakistani Rs in to euro\n"
             << "Press 2 :Call function which convert euro to yen\n"
             << "Press 3 :Dollar in to euro\n"
             << "Press 4 :Pakistani Rs to dollar\n"
             << "Press 5 :Dollar in to yen \n";
        cout << ">> ";
        cin >> choice;
    } while(!(choice > 0 && choice < 6));

    cout << "Enter amount: ";
    cin >> amount;

    switch (choice)
    {
    case 1:
        cout << "Converted amount is: " << rupeesToEuro(amount) << '\n';
        break;

    case 2:
        cout << "Converted amount is: " << euroToYen(amount) << '\n';
        break;

    case 3:
        cout << "Converted amount is: " << dollarToEuro(amount) << '\n';
        break;

    case 4:
        cout << "Converted amount is: " << rupeesToDollar(amount) << '\n';
        break;

    case 5:
        cout << "Converted amount is: " << dollarToYen(amount) << '\n';
        break;

    default:
        break;
    }

}
```

```
int main(void) {

    currencyConverterMenu();

    return EXIT_SUCCESS;
}
```

OUTPUT:



Part B

Write a program which takes 10 student marks from user. The system shall allow user to use three functionalities i.e. insert (), delete(), update(). All the student marks are inserted in an array via functions.

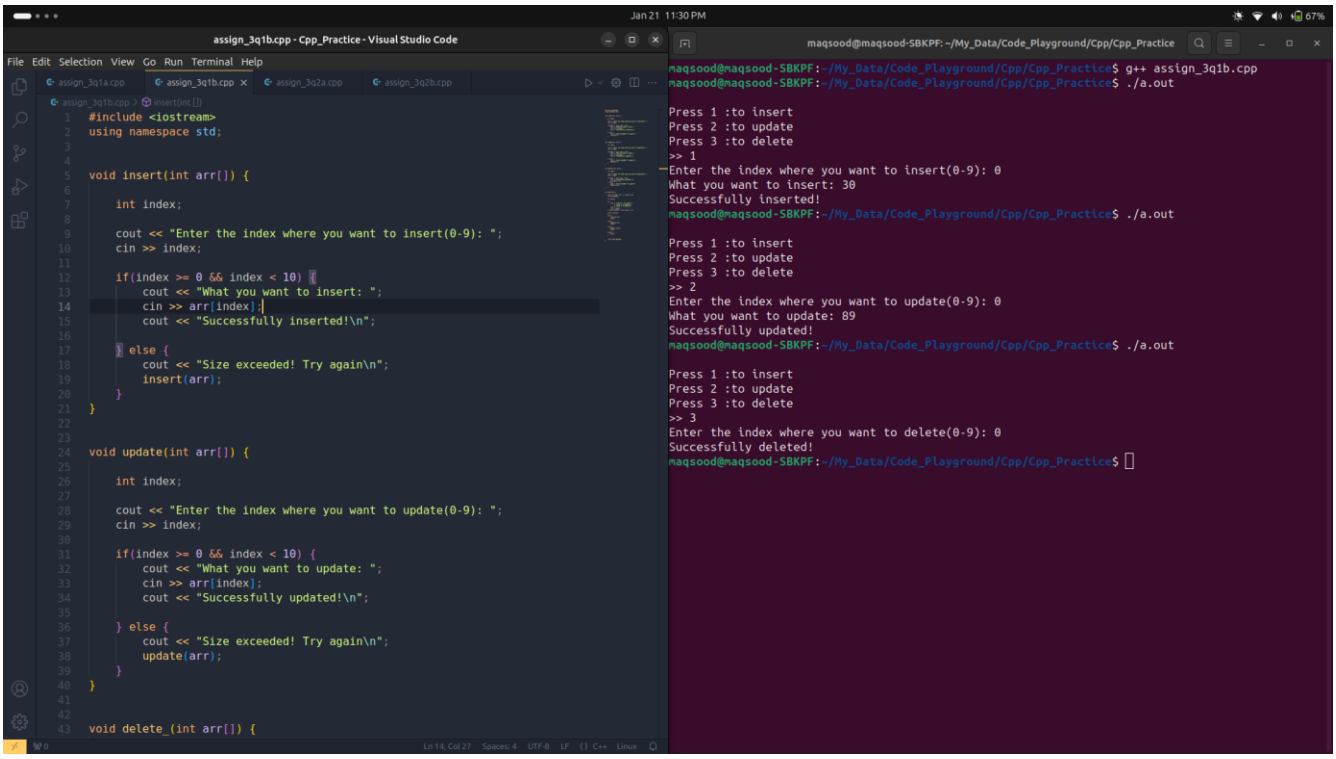
Source Code:

```
#include <iostream>
using namespace std;


void insert(int arr[]) {

   int index;

   cout << "Enter the index where you want to insert(0-9): ";
   cin >> index;

   if(index >= 0 && index < 10) {
      cout << "What you want to insert: ";
      cin >> arr[index];
      cout << "Successfully inserted!\n";

   } else {
      cout << "Size exceeded! Try again\n";
      insert(arr);
   }
}


void update(int arr[]) {

   int index;

   cout << "Enter the index where you want to update(0-9): ";
   cin >> index;

   if(index >= 0 && index < 10) {
      cout << "What you want to update: ";
      cin >> arr[index];
      cout << "Successfully updated!\n";

   } else {
```

```
        cout << "Size exceeded! Try again\n";
        update(arr);
    }
}


void delete_(int arr[]) {

    int index;

    cout << "Enter the index where you want to delete(0-9): ";
    cin >> index;

    if(index >= 0 && index < 10) {
        cout << "Successfully deleted!\n";
        arr[index] = 0;
    } else {
        cout << "Size exceeded! Try again\n";
        delete_(arr);
    }
}


int main(void) {

    const int SIZE = 10; // constant size
    int array[SIZE];

    int choice;

    do {
        system("clear"); // clears the screen
        cout << "\nPress 1 :to insert\n"
             << "Press 2 :to update\n"
             << "Press 3 :to delete\n";
        cout << ">> ";
        cin >> choice;
    } while(!(choice > 0 && choice < 3));

    switch (choice)
    {
    case 1:
        insert(array);
        break;

    case 2:
        update(array);
        break;
```

```
    case 3:
        delete_(array);
        break;

    default:
        break;
    }


    return EXIT_SUCCESS;
}
```

OUTPUT:



Question 02[CLO3: C3][20]

Part A

Write a C++ program that creates a structure called course that contains two members

- ➔ **Name (string)**
- ➔ **Credit hr(int)**

After that creates a structure called student that contains the following data members.

- ➔ **Name(string)**
- ➔ **Age(int)**
- ➔ **Cms(int)**
- ➔ **Semester(int)**
- ➔ **Course (course)**

Take the data member's value from the user and display the record of the 3 students.

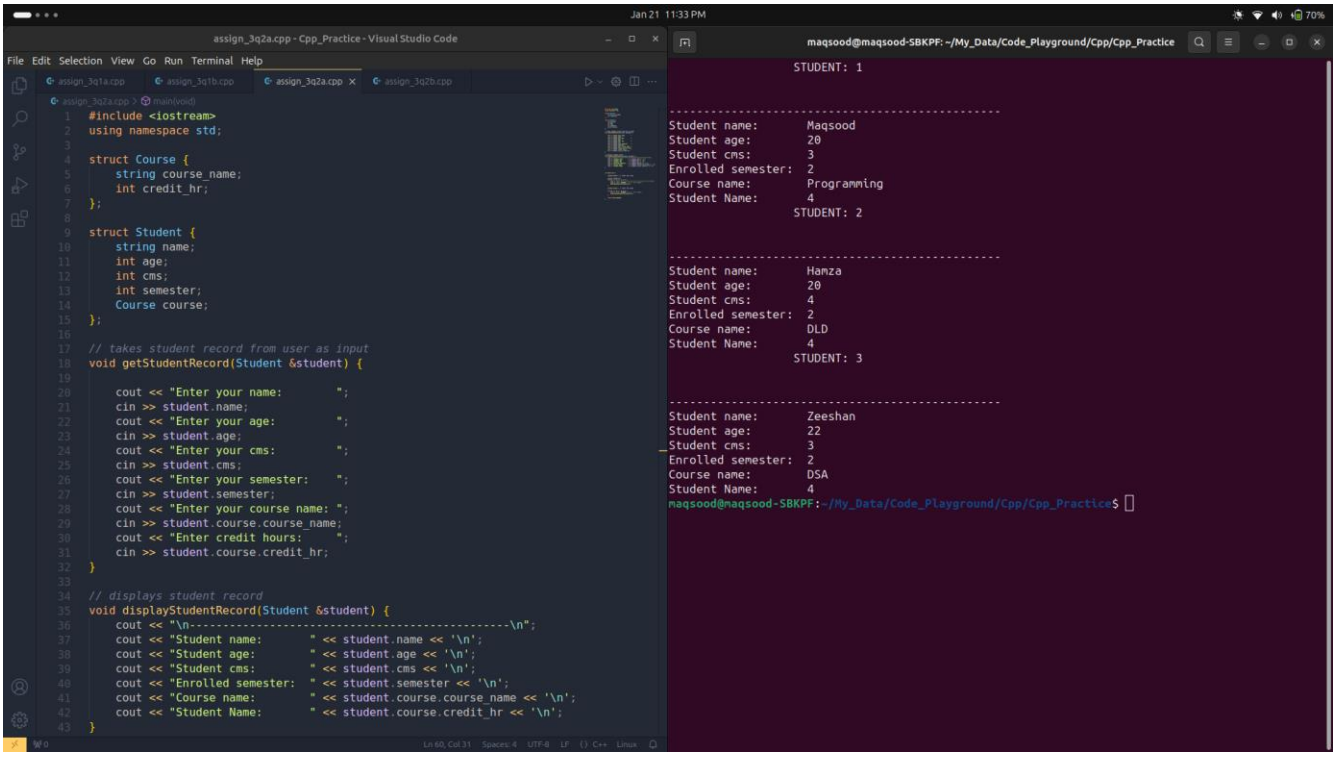
Source Code:

```
#include <iostream>
using namespace std;

struct Course {
   string course_name;
   int credit_hr;
};

struct Student {
   string name;
   int age;
   int cms;
   int semester;
   Course course;
};
```

```
// takes student record from user as input
void getStudentRecord(Student &student) {

    cout << "Enter your name:        ";
    cin >> student.name;
    cout << "Enter your age:         ";
    cin >> student.age;
    cout << "Enter your cms:         ";
    cin >> student.cms;
    cout << "Enter your semester:    ";
    cin >> student.semester;
    cout << "Enter your course name: ";
    cin >> student.course.course_name;
    cout << "Enter credit hours:     ";
    cin >> student.course.credit_hr;
}

// displays student record
void displayStudentRecord(Student &student) {
    cout << "\n-----------------------------------------------\n";
    cout << "Student name:       " << student.name << '\n';
    cout << "Student age:        " << student.age << '\n';
    cout << "Student cms:        " << student.cms << '\n';
    cout << "Enrolled semester:  " << student.semester << '\n';
    cout << "Course name:        " << student.course.course_name << '\n';
    cout << "Student Name:       " << student.course.credit_hr << '\n';
}



int main(void) {

    system("clear"); // clears the screen

    Student student[3];
    for(int i = 0; i < 3; i++) {
        cout << "\n-----------------------------------------------\n";
        cout << "\t\t  STUDENT: " << i + 1 << "\n\n";
        getStudentRecord(student[i]);
    }

    system("clear"); // clears the screen
```

```
    for(int i = 0; i < 3; i++) {
        cout << "\t\t  STUDENT: " << i + 1 << "\n\n";
        displayStudentRecord(student[i]);
    }

    return EXIT_SUCCESS;
}
```

OUTPUT:



Part B

Write a C++ program that creates a structure to manage a record of employee. Employee has 4 attribute.

- **➔ Employee id**
- **➔ Employee Name**
- **➔ Employee age**
- **➔ Employee salary**

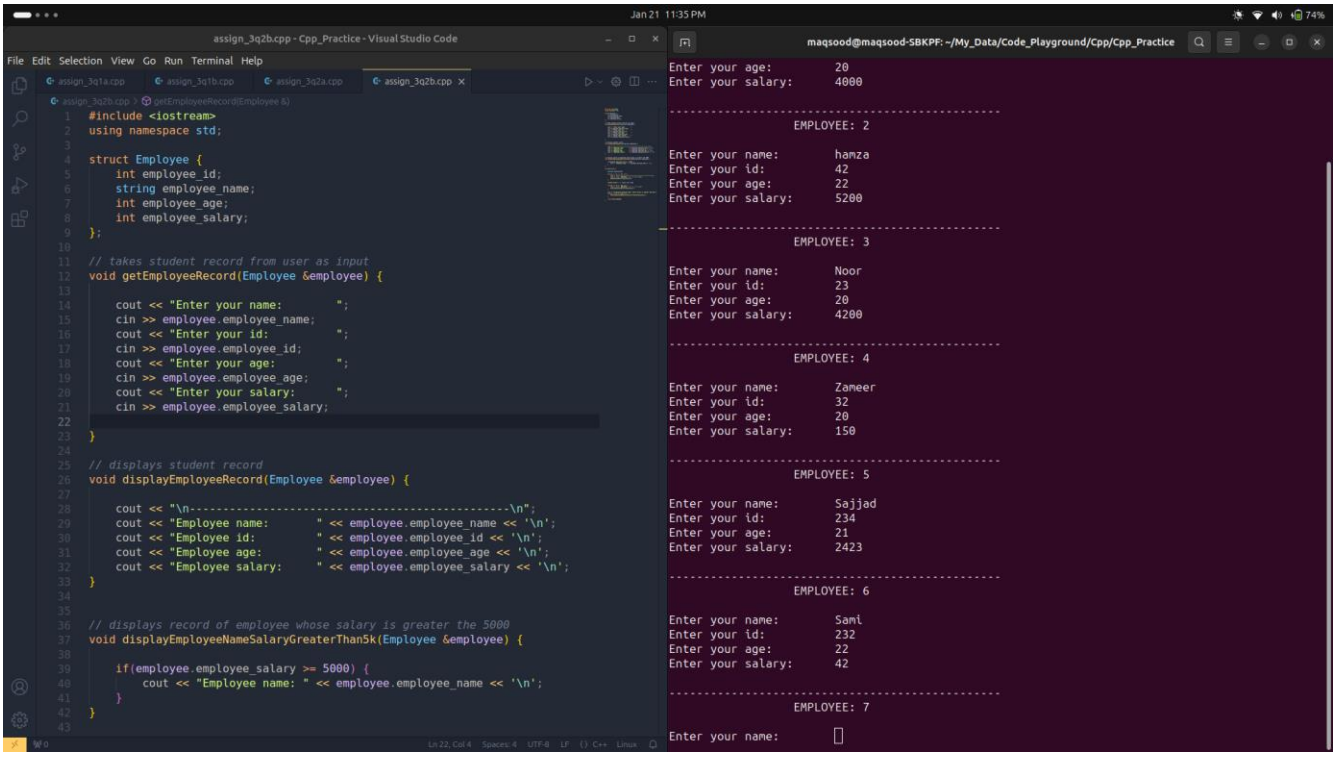
Make a structure of employee having provided attributes and store 10 employee data and display it. Also display the name of those employee whose salary is greater than 5000.

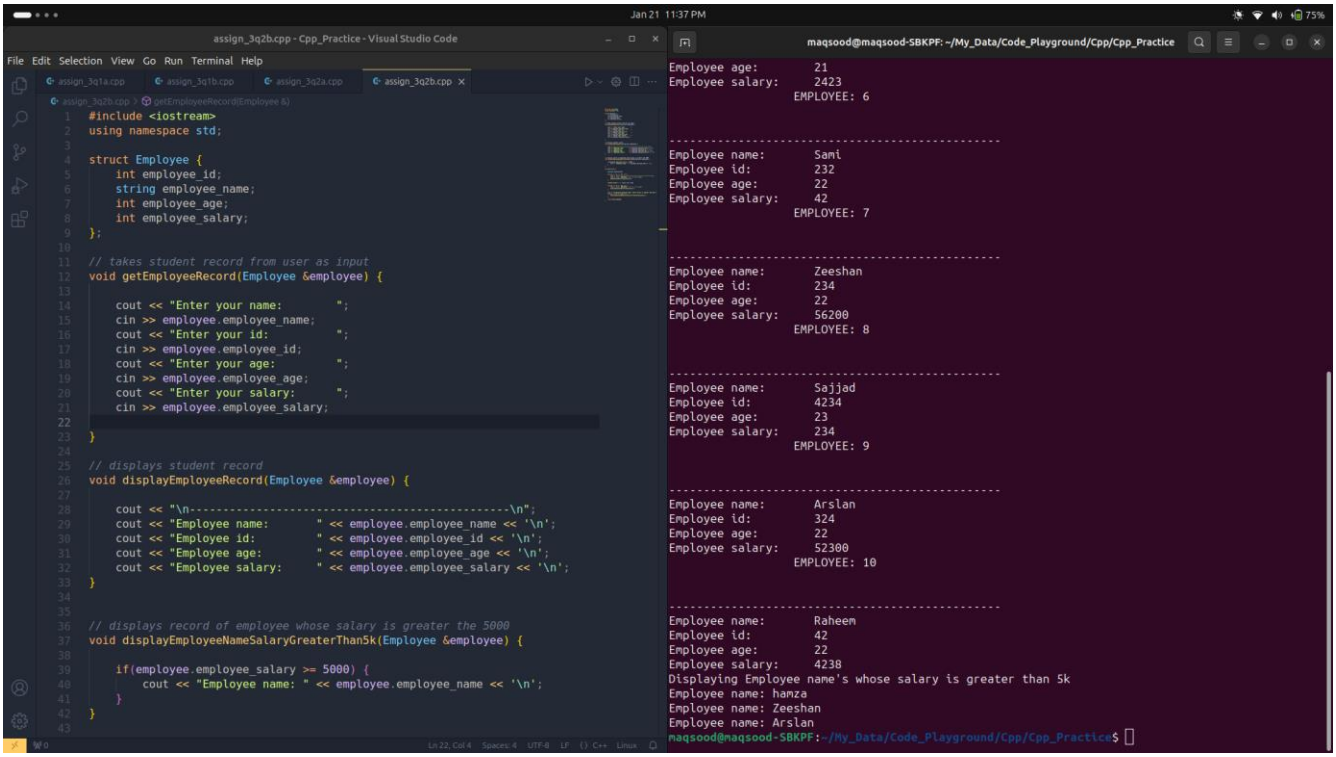
Source Code:

```
#include <iostream>
using namespace std;

struct Employee {
   int employee_id;
   string employee_name;
   int employee_age;
   int employee_salary;
};

// takes student record from user as input
void getEmployeeRecord(Employee &employee) {

   cout << "Enter your name:       ";
   cin >> employee.employee_name;
   cout << "Enter your id:         ";
   cin >> employee.employee_id;
   cout << "Enter your age:        ";
   cin >> employee.employee_age;
   cout << "Enter your salary:     ";
   cin >> employee.employee_salary;

}

// displays student record
```

```
void displayEmployeeRecord(Employee &employee) {

   cout << "\n-----------------------------------------------\n";
   cout << "Employee name:       " << employee.employee_name << '\n';
   cout << "Employee id:         " << employee.employee_id << '\n';
   cout << "Employee age:        " << employee.employee_age << '\n';
   cout << "Employee salary:     " << employee.employee_salary << '\n';
}


// displays record of employee whose salary is greater the 5000
void displayEmployeeNameSalaryGreaterThan5k(Employee &employee) {

   if(employee.employee_salary >= 5000) {
      cout << "Employee name: " << employee.employee_name << '\n';
   }
}

int main(void) {
   Employee employee[10];

   for(int i = 0; i < 10; i++) {
      cout << "\n-----------------------------------------------\n";
      cout << "\t\t  EMPLOYEE: " << i + 1 << "\n\n";
      getEmployeeRecord(employee[i]);
   }

   system("clear"); // clears the screen

   for(int i = 0; i < 10; i++) {
      cout << "\t\t  EMPLOYEE: " << i + 1 << "\n\n";
      displayEmployeeRecord(employee[i]);
   }

   cout << "Displaying Employee name's whose salary is greater than 5k\n";
   for(int i = 0; i < 10; i++) {
      displayEmployeeNameSalaryGreaterThan5k(employee[i]);
   }
   return EXIT_SUCCESS;
}
```

OUTPUT:


Jan 21 11:35 PM
assign_3q2b.cpp - Cpp_Practice - Visual Studio Code
File Edit Selection View Go Run Terminal Help
assign_3q1a.cpp
assign_3q1b.cpp
assign_3q2a.cpp
assign_3q2b.cpp
#include <iostream>
using namespace std;
struct Employee {
int employee_id;
string employee_name;
int employee_age;
int employee_salary;
};
// takes student record from user as input
void getEmployeeRecord(Employee &employee) {
cout << "Enter your name: ";
cin >> employee.employee_name;
cout << "Enter your id: ";
cin >> employee.employee_id;
cout << "Enter your age: ";
cin >> employee.employee_age;
cout << "Enter your salary: ";
cin >> employee.employee_salary;
}
// displays student record
void displayEmployeeRecord(Employee &employee) {
cout << "\n---\n";
cout << "Employee name: " << employee.employee_name << '\n';
cout << "Employee id: " << employee.employee_id << '\n';
cout << "Employee age: " << employee.employee_age << '\n';
cout << "Employee salary: " << employee.employee_salary << '\n';
}
// displays record of employee whose salary is greater the 5000
void displayEmployeeNameSalaryGreaterThan5k(Employee &employee) {
if(employee.employee_salary >= 5000) {
cout << "Employee name: " << employee.employee_name << '\n';
}
}
maqsood@maqsood-SBKPF: ~/My_Data/Code_Playground/Cpp/Cpp_Practice
Enter your age: 20
Enter your salary: 4000
EMPLOYEE: 2
Enter your name: hamza
Enter your id: 42
Enter your age: 22
Enter your salary: 5200
EMPLOYEE: 3
Enter your name: Noor
Enter your id: 23
Enter your age: 20
Enter your salary: 4200
EMPLOYEE: 4
Enter your name: Zameer
Enter your id: 32
Enter your age: 20
Enter your salary: 150
EMPLOYEE: 5
Enter your name: Sajjad
Enter your id: 234
Enter your age: 21
Enter your salary: 2423
EMPLOYEE: 6
Enter your name: Sami
Enter your id: 232
Enter your age: 22
Enter your salary: 42
EMPLOYEE: 7
Enter your name:


Jan 21 11:37 PM
assign_3q2b.cpp - Cpp_Practice - Visual Studio Code
File Edit Selection View Go Run Terminal Help
#include <iostream>
using namespace std;
struct Employee {
int employee_id;
string employee_name;
int employee_age;
int employee_salary;
};
// takes student record from user as input
void getEmployeeRecord(Employee &employee) {
cout << "Enter your name: ";
cin >> employee.employee_name;
cout << "Enter your id: ";
cin >> employee.employee_id;
cout << "Enter your age: ";
cin >> employee.employee_age;
cout << "Enter your salary: ";
cin >> employee.employee_salary;
}
// displays student record
void displayEmployeeRecord(Employee &employee) {
cout << "\n---\n";
cout << "Employee name: " << employee.employee_name << '\n';
cout << "Employee id: " << employee.employee_id << '\n';
cout << "Employee age: " << employee.employee_age << '\n';
cout << "Employee salary: " << employee.employee_salary << '\n';
}
// displays record of employee whose salary is greater the 5000
void displayEmployeeNameSalaryGreaterThan5k(Employee &employee) {
if(employee.employee_salary >= 5000) {
cout << "Employee name: " << employee.employee_name << '\n';
}
}
maqsood@maqsood-SBKPF: ~/My_Data/Code_Playground/Cpp/Cpp_Practice
Employee age: 21
Employee salary: 2423
EMPLOYEE: 6
Employee name: Sami
Employee id: 232
Employee age: 22
Employee salary: 42
EMPLOYEE: 7
Employee name: Zeeshan
Employee id: 234
Employee age: 22
Employee salary: 56200
EMPLOYEE: 8
Employee name: Sajjad
Employee id: 4234
Employee age: 23
Employee salary: 234
EMPLOYEE: 9
Employee name: Arslan
Employee id: 324
Employee age: 22
Employee salary: 52300
EMPLOYEE: 10
Employee name: Raheem
Employee id: 42
Employee age: 22
Employee salary: 4238
Displaying Employee name's whose salary is greater than 5k
Employee name: hamza
Employee name: Zeeshan
Employee name: Arslan
maqsood@maqsood-SBKPF:~/My_Data/Code_Playground/Cpp/Cpp_Practice$