Programming Fundamentals (CSC-115) Instructor: Hamna Anwar

Assignment 3 solution Max Marks:15

Due Date: 16th May 2024

Name: DUA SHAKEEL

Roll number: F2023-134

A1: #include <iostream>  
using namespace std;  
void calGrade(int marks, char &grade) {  
 if (marks >= 80) {  
 grade = 'A';  
 } else if (marks >= 65) {  
 grade = 'B';  
 } else if (marks >= 50) {  
 grade = 'C';  
 } else {  
 grade = 'F';  
 }  
}  
  
int main() {  
 int marks;  
 char grade;  
 char repeat;  
  
 do {  
 cout << "Enter marks (0-100, -1 to exit): ";  
 cin >> marks;  
  
 if (marks == -1) {  
 break;  
 }  
  
 if (marks < 0 || marks > 100) {  
 cout << "Invalid marks entered. Please enter marks between 0 and 100." << std::endl;  
 continue;  
 }  
  
 calGrade(marks, grade);  
 cout << "Grade: " << grade << std::endl;  
 cout << "Do you want to calculate grade for another marks? (Y/N): ";  
 cin >> repeat;  
 } while (repeat == 'Y' || repeat == 'y');  
  
 cout << "Program Exit." << std::endl;  
 return 0;  
}

A2: #include <iostream>  
using namespace std;  
  
double calCharges(int days, double dailyRate, double medCharges, double serviceCharges);  
double calCharges(double medCharges, double serviceCharges);  
  
int main() {  
 int days;  
 double dailyRate, medCharges, serviceCharges;  
 char choice;  
  
 cout << "Enter 'i' for in-patient or 'o' for out-patient: ";  
 cin >> choice;  
  
 if (choice == 'i') {  
 cout << "Enter number of days spent in hospital: ";  
 cin >> days;  
 cout << "Enter daily rate: ";  
 cin >> dailyRate;  
 cout << "Enter medication charges: ";  
 cin >> medCharges;  
 cout << "Enter charges for hospital services: ";  
 cin >> serviceCharges;  
  
 cout << "Total Charges: $" << calCharges(days, dailyRate, medCharges, serviceCharges) << endl;  
 } else if (choice == 'o') {  
 cout << "Enter medication charges: ";  
 cin >> medCharges;  
 cout << "Enter charges for hospital services: ";  
 cin >> serviceCharges;  
  
 cout << "Total: $" << calCharges(medCharges, serviceCharges) << endl;  
 } else {  
 cout << "Invalid choice." << endl;  
 }  
  
 return 0;  
}  
  
double calCharges(int days, double dailyRate, double medCharges, double serviceCharges) {  
 return days \* dailyRate + medCharges + serviceCharges;  
}  
  
double calCharges(double medCharges, double serviceCharges) {  
 return medCharges + serviceCharges;  
}

A3: #include <iostream>  
using namespace std;  
  
void DistinctEle(int arr[], int n, int DistinctArr[]) {  
 int k = 0;  
 for (int i = 0; i < n; i++) {  
 bool isDistinct = true;  
 for (int j = 0; j < i; j++) {  
 if (arr[i] == arr[j]) {  
 isDistinct = false;  
 break;  
 }  
 }  
 if (isDistinct) {  
 DistinctArr[k++] = arr[i];  
 }  
 }  
}  
  
void displayDistinctElements (int DistinctArr[], int k) {  
 cout << "Distinct Elements: ";  
 for (int i = 0; i < k; i++) {  
 cout << DistinctArr[i] << " ";  
 }  
 cout << endl;  
}  
  
int main() {  
 int arr[10];  
 int DistinctArr[10];  
 int n;  
  
 cout << "Enter the number of elements (up to 10): ";  
 cin >> n;  
  
 cout << "Enter " << n << " elements: ";  
 for (int i = 0; i < n; i++) {  
 cin >> arr[i];  
 }  
  
 DistinctEle (arr, n, DistinctArr);  
 displayDistinctElements(DistinctArr, n);  
  
 return 0;  
}

A4: #include <iostream>  
#include <set>  
using namespace std;  
  
void validateData(int& size) {  
 cout << "Enter the size of the array (positive integer): ";  
 cin >> size;  
 while (size <= 0) {  
 cout << "Invalid input. Enter the size of the array (positive integer): ";  
 cin >> size;  
 }  
}  
  
void fillArray(int arr[], int size) {  
 cout << "Enter " << size << " elements: ";  
 for (int i = 0; i < size; i++) {  
 cin >> arr[i];  
 }  
}  
  
bool Duplicates(int arr[], int size) {  
 set<int> uniqueElements;  
 for (int i = 0; i < size; i++) {  
 uniqueElements.insert(arr[i]);  
 }  
 return uniqueElements.size() < size;  
}  
  
int main() {  
 int size;  
 int arr[10];  
  
 validateData(size);  
 fillArray(arr, size);  
  
 if (Duplicates(arr, size)) {  
 cout << "Duplicates found" << endl;  
 } else {  
 cout << "Duplicates not found" << endl;  
 }  
  
 return 0;  
}