**/\*1 To calculate the area of circle, rectangle and triangle using function overloading\*/**

G MAHESH – PES1PG23CA331 – ARC ASSI

#include<iostream>

using namespace std;

class Area

{

public:

    void area(int r)

    {

        cout << "Area of circle is " << 3.14 \* r \* r << endl;

    }

    void area(int l, int b)

    {

        cout << "Area of rectangle is " << l \* b << endl;

    }

    void area(double b, double h)

    {

        cout << "Area of triangle is " << 0.5 \* b \* h << endl;

    }

};

int main()

{

    Area a;

    a.area(5);

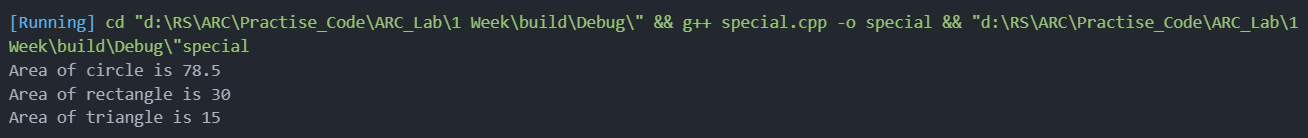
    a.area(5, 6);

    a.area(5.0, 6.0);  // Use double values for the triangle area

    return 0;

}

**Output:**



**/\*2 Prints the English name of an integer from 1 to 9\*/**

#include<iostream>

using namespace std;

class Number

{

public:

    void print(int n)

    {

        switch (n)

        {

        case 1:

            cout << "One" << endl;

            break;

        case 2:

            cout << "Two" << endl;

            break;

        case 3:

            cout << "Three" << endl;

            break;

        case 4:

            cout << "Four" << endl;

            break;

        case 5:

            cout << "Five" << endl;

            break;

        case 6:

            cout << "Six" << endl;

            break;

        case 7:

            cout << "Seven" << endl;

            break;

        case 8:

            cout << "Eight" << endl;

            break;

        case 9:

            cout << "Nine" << endl;

            break;

        default:

            cout << "Invalid input" << endl;

        }

    }

};

int main()

{

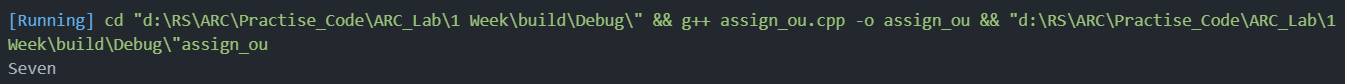
    Number n;

    n.print(5);

    return 0;

}

**Output:**



**/\*3 Find the index of the largest number and smallest number in an array\*/**

#include<iostream>

#include<vector>

using namespace std;

int main() {

    vector<int> arr = {12, 34, 7, 89, 45, 22};

    int maxIndex = 0, minIndex = 0;

    int maxNum = arr[0], minNum = arr[0];

    for(int i = 1; i < arr.size(); i++) {

        if(arr[i] > maxNum) {

            maxNum = arr[i];

            maxIndex = i;

        }

        if(arr[i] < minNum) {

            minNum = arr[i];

            minIndex = i;

        }

    }

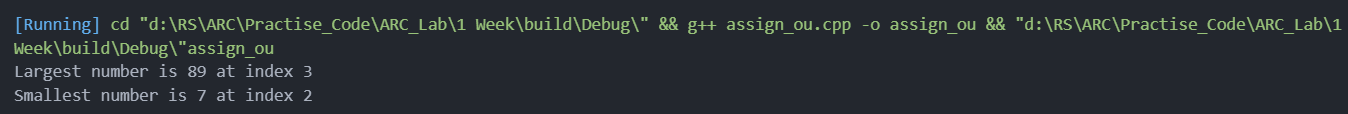
    cout << "Largest number is " << maxNum << " at index " << maxIndex << endl;

    cout << "Smallest number is " << minNum << " at index " << minIndex << endl;

    return 0;

}

**Output:**



**/\*4.Write a program to eliminate duplicates from an array\*/**

#include<iostream>

using namespace std;

int main ()

{

    int A[10], B[10], n, i, j, k = 0;

    cout << "Enter size of array : ";

    cin >> n;

    cout << "Enter elements of array : ";

    for (i = 0; i < n; i++)

        cin >> A[i];

    for (i = 0; i < n; i++)

    {

        for (j = 0; j < k; j++)

        {

            if (A[i] == B[j])

                break;

        }

        if (j == k)

        {

            B[k] = A[i];

            k++;

        }

    }

    cout << "Repeated elements after deletion : ";

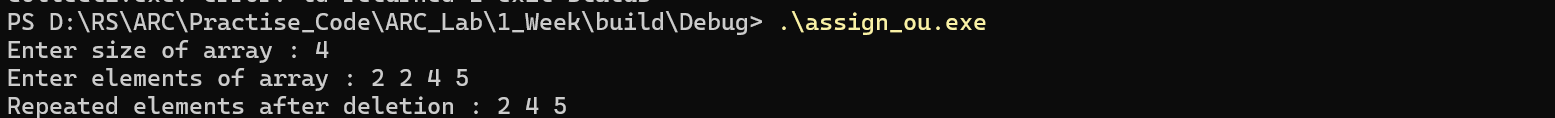
    for (i = 0; i < k; i++)

        cout << B[i] << " ";

    return 0;

}

**Output:**

****

**/\*5.    A library wants to analyse book ratings given by readers.**

**Requirements:**

**Store ratings of M books in an array (values between 1 to 5).**

**Implement the following functionalities:**

**Calculate the average rating.**

**Count how many books have a rating of 4 or 5.**

**Find the book with the highest and lowest rating.**

**Sort the ratings in descending order.**

**\*/**

#include<iostream>

#include<vector>

#include<algorithm>

using namespace std;

double calculateAverageRating(vector<int> ratings) {

    int sum = 0;

    for(int rating : ratings) {

        sum += rating;

    }

    return (double)sum / ratings.size();

}

int countRatingsAboveFourOrFive(vector<int> ratings) {

    int count = 0;

    for(int rating : ratings) {

        if(rating >= 4) {

            count++;

        }

    }

    return count;

}

int main() {

    vector<int> ratings = {4, 3, 5, 4, 5, 2, 3, 4, 5, 5};

    double averageRating = calculateAverageRating(ratings);

    int booksWithRatingAboveFourOrFive = countRatingsAboveFourOrFive(ratings);

    cout << "Average rating: " << averageRating << endl;

    cout << "Books with rating 4 or 5: " << booksWithRatingAboveFourOrFive << endl;

    int highestRating = \*max\_element(ratings.begin(), ratings.end());

    int lowestRating = \*min\_element(ratings.begin(), ratings.end());

    cout << "Highest rating: " << highestRating << endl;

    cout << "Lowest rating: " << lowestRating << endl;

    sort(ratings.begin(), ratings.end(), greater<int>());

    cout << "Sorted ratings in descending order: ";

    for(int rating : ratings) {

        cout << rating << " ";

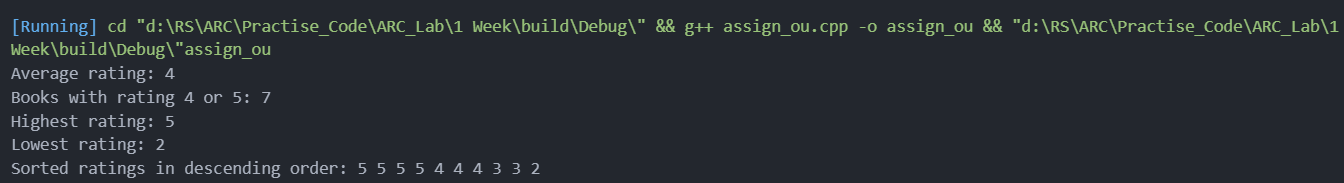
    }

    cout << endl;

    return 0;

    }

**Output:**



**/\*6.    Temperature Data Analysis**

**You are developing a weather tracking system that records daily temperatures.**

**Requirements:**

**Store weekly temperature data in an array.**

**Implement the following functionalities:**

**Find the average temperature of the week.**

**Identify the hottest and coldest day.**

**Find the number of days when the temperature was above 30°C.**

**Sort temperatures in descending order.**

**\*/**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

double calculateAverageTemperature(vector<double> temperatures) {

    double sum = 0;

    for(double temperature : temperatures) {

        sum += temperature;

    }

    return sum / temperatures.size();

}

pair<double, double> findHottestAndColdestDay(vector<double> temperatures) {

    double hottestTemperature = temperatures[0];

    double coldestTemperature = temperatures[0];

    for(double temperature : temperatures) {

        if(temperature > hottestTemperature) {

            hottestTemperature = temperature;

        }

        if(temperature < coldestTemperature) {

            coldestTemperature = temperature;

        }

    }

    return {hottestTemperature, coldestTemperature};

}

int countDaysAboveThirty(vector<double> temperatures) {

    int count = 0;

    for(double temperature : temperatures) {

        if(temperature > 30) {

            count++;

        }

    }

    return count;

    }

int main() {

    vector<double> temperatures = {25.5, 28.2, 29.8, 27.6, 26.9, 30.1, 31.4};

    double averageTemperature = calculateAverageTemperature(temperatures);

    pair<double, double> hottestAndColdestDay = findHottestAndColdestDay(temperatures);

    cout << "Average temperature: " << averageTemperature << endl;

    cout << "Hottest day: " << hottestAndColdestDay.first << endl;

    cout << "Coldest day: " << hottestAndColdestDay.second << endl;

    cout << "Number of days above 30°C: " << countDaysAboveThirty(temperatures) << endl;

    sort(temperatures.begin(), temperatures.end(), greater<double>());

    cout << "Sorted temperatures in descending order: ";

    for(double temperature : temperatures) {

        cout << temperature << " ";

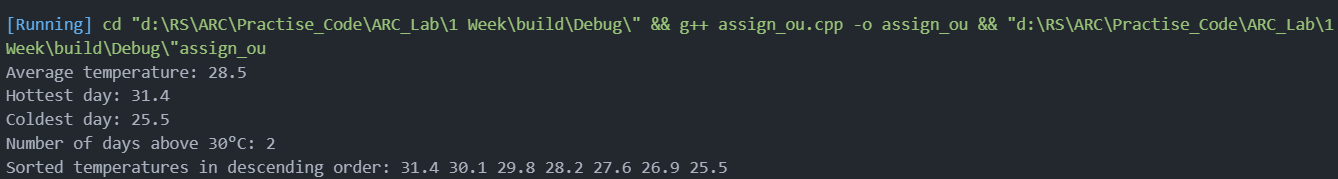
    }

    cout << endl;

    return 0;

    }

**Output:**



**/\*7.    Create a class player with following data members name, age , runs, highest , lowest, test, average. Write member function for each of the following: (i) to get data (ii) to display data (iii) to calculate average of player.\*/**

#include<iostream>

using namespace std;

class player {

    public:

    string name;

    int age;

    int runs;

    int highest;

    int lowest;

    int test;

    double average;

    void getdata(){

        cout<<"Enter the name of the player: ";

        cin>>name;

        cout<<"Enter the age of the player: ";

        cin>>age;

        cout<<"Enter the runs of the player: ";

        cin>>runs;

        cout<<"Enter the highest score of the player: ";

        cin>>highest;

        cout<<"Enter the lowest score of the player: ";

        cin>>lowest;

        cout<<"Enter the number of test matches played by the player: ";

        cin>>test;

        average = runs/test;

        cout<<endl;

        cout<<"Player Details: "<<endl;

        cout<<"Name: "<<name<<endl;

        cout<<"Age: "<<age<<endl;

        cout<<"Runs: "<<runs<<endl;

        cout<<"Highest Score: "<<highest<<endl;

        cout<<"Lowest Score: "<<lowest<<endl;

        cout<<"Number of Test Matches: "<<test<<endl;

        cout<<"Average Score: "<<average<<endl;

        cout<<endl;

        }

    };

    int main(){

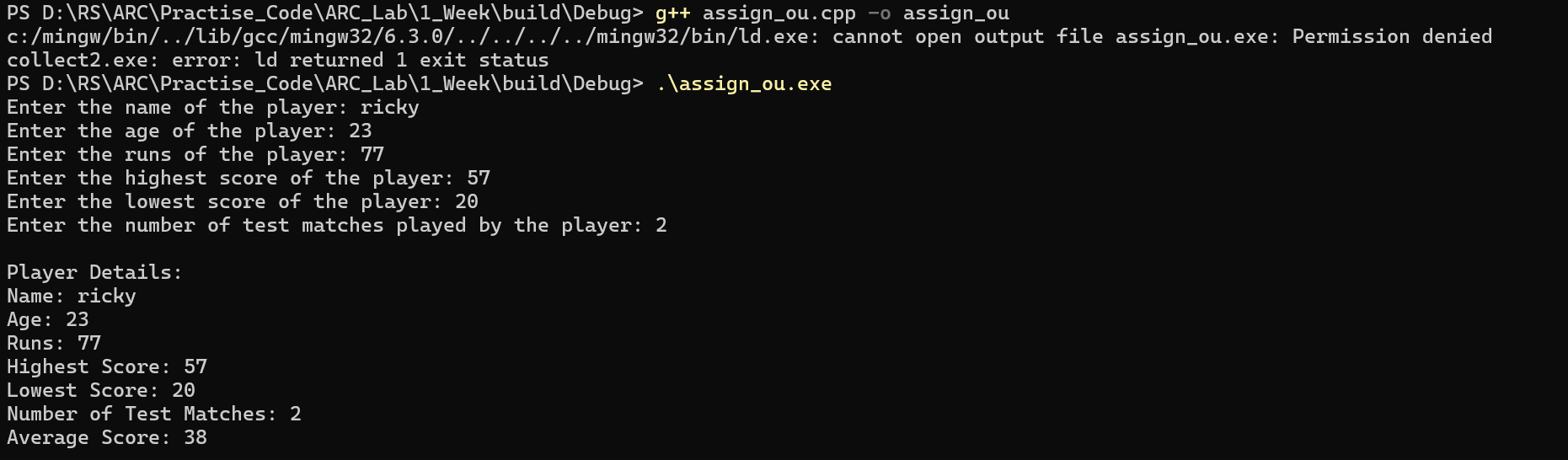
        player p;

        p.getdata();

        return 0;

        }

**Output:**



**/\*8.    C++ Program for Calculating Labor Wage Based on Hours Worked in a Day using class and object with the following member variables employee name, emp id , no of hours,  total wages  and member function  get data to read the inputs Calculate the wage(), display()**

**Hours Worked    Rate**

**the first 8 hours   50**

**next 4 hours    10/hr**

**next 4 hours    20/hr**

**next 4 hours    25/hr**

**next 4 hours    40/hr**

**\*/**

#include<iostream>

using namespace std;

class labor{

    public:

    string name;

    int id;

    int hours;

    int wage;

    void getdata(){

        cout<<"Enter the name of the employee: ";

        cin>>name;

        cout<<"Enter the employee id: ";

        cin>>id;

        cout<<"Enter the number of hours worked: ";

        cin>>hours;

        cout<<endl;

        }

    void calculate(){

        if(hours<=8){

            wage = hours\*50;

            }

        else if(hours>8 && hours<=12){

            wage = 8\*50 + (hours-8)\*10;

            }

        else if(hours>12 && hours<=16){

            wage = 8\*50 + 4\*10 + (hours-12)\*20;

            }

        else if(hours>16 && hours<=20){

            wage = 8\*50 + 4\*10 + 4\*20 + (hours-16)\*25;

            }

        else{

            wage = 8\*50 + 4\*10 + 4\*20 + 4\*25 + (hours-20)\*40;

            }

        }

    void display(){

        cout<<"Employee Details: "<<endl;

        cout<<"Name: "<<name<<endl;

        cout<<"Employee ID: "<<id<<endl;

        cout<<"Number of hours worked: "<<hours<<endl;

        cout<<"Total wage: "<<wage<<endl;

        }

    };

    int main(){

        labor l;

        l.getdata();

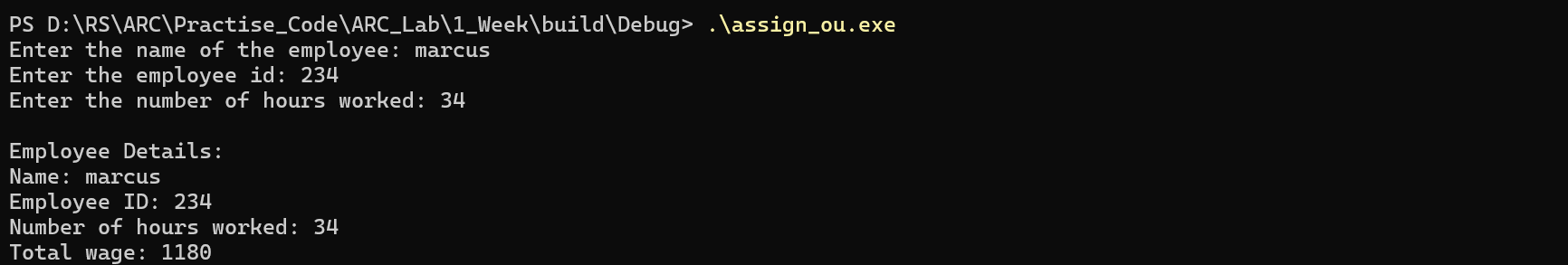
        l.calculate();

        l.display();

        return 0;

        }

**Output:**



**/\*9.    Suppose you have a Piggie Bank with an initial amount of $50 and you have to add some more amount to it. Create a class 'AddAmount' with a data member named 'amount' with an initial value of $50. Now make two constructors of this class as follows:**

**1 - without any parameter - no amount will be added to the Piggie Bank**

**2 - having a parameter which is the amount that will be added to the Piggie Bank**

**Create an object of the 'AddAmount' class and display the final amount in the Piggie Bank.**

**\*/**

#include<iostream>

using namespace std;

class AddAmount{

    public:

    int amount = 50;

    AddAmount(){

        cout<<"Amount in the Piggie Bank: "<<amount<<endl;

        }

    AddAmount(int a){

        amount = amount + a;

        cout<<"Amount in the Piggie Bank: "<<amount<<endl;

        }

    };

    int main(){

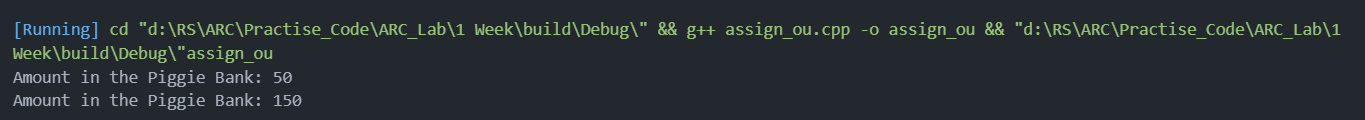
        AddAmount a1;

        AddAmount a2(100);

        return 0;

    }

**Output:**



**/\* 10. Create a class Time with the following data members int m, h, s. Write a constructor and copy constructor to initialize the objects and member function for(I ) to convert sec in minutes and hours (ii)to display data.\*/**

#include<iostream>

using namespace std;

class Time{

    public:

    int m,h,s;

    Time(int a, int b, int c){

        h = a;

        m = b;

        s = c;

        }

    Time(const Time &t){

        h = t.h;

        m = t.m;

        s = t.s;

        }

    void convert(){

        m = s/60;

        s = s%60;

        h = h + m/60;

        m = m%60;

        }

    void display(){

        cout<<"Hours: "<<h<<endl;

        cout<<"Minutes: "<<m<<endl;

        cout<<"Seconds: "<<s<<endl;

        }

    };

    int main(){

        Time t1(1,2,3);

        Time t2(t1);

        t1.convert();

        t2.convert();

        t1.display();

        t2.display();

        return 0;

    }

**Output:**

