

Submitted by : Spogmai jan

Roll no : 2430 - 0081

Submitted to : sir bilal khan

Subject : oop lab

**Assignment :06**

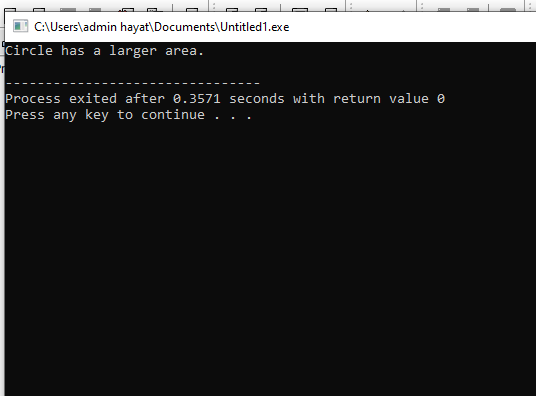
**Task 1**

Define two classes, Rectangle and Circle, each having a private data member representing an area. Create a friend function that compares the area of a Rectangle object with that of a Circle object and outputs which area is larger.

**CODE :**

|  |
| --- |
| #include <iostream>  using namespace std;  class Circle;  class Rectangle {  private:  double area;  public:  Rectangle(double length, double width) {  area = length \* width;  }  friend void compareAreas(const Rectangle &r, const Circle &c);  };  class Circle {  private:  double area;  public:  Circle(double radius) {  area = 3.14159 \* radius \* radius;  }  friend void compareAreas(const Rectangle &r, const Circle &c);  };  void compareAreas(const Rectangle &r, const Circle &c) {  if (r.area > c.area)  cout << "Rectangle has a larger area." << endl;  else if (r.area < c.area)  cout << "Circle has a larger area." << endl;  else  cout << "Both have the same area." << endl;  }  int main() {  Rectangle rect(10, 5);  Circle circ(5);  compareAreas(rect, circ);  return 0;  } |

**OUTPUT:**



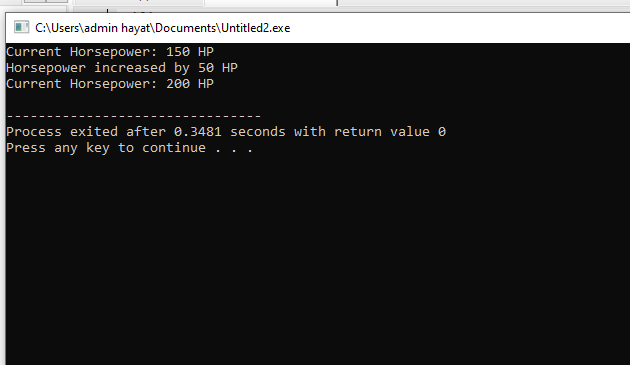
**Task2**

Write a C++ program with two classes, Engine and Car. Make Engine a friend class of Car so that it can access and modify the private data members of Car. Implement a function in Engine that increases the horsepower of Car and demonstrates the use of a friend class to access and modify Car’s private data.

**CODE:**

|  |
| --- |
| #include <iostream>  using namespace std;  class Car; // Forward declaration  class Engine {  public:  void increaseHorsepower(Car &c, int hp);  };  class Car {  private:  int horsepower;  public:  Car(int hp) : horsepower(hp) {}  void showHorsepower() {  cout << "Current Horsepower: " << horsepower << " HP" << endl;  }  friend class Engine; // Making Engine a friend of Car  };  void Engine::increaseHorsepower(Car &c, int hp) {  c.horsepower += hp;  cout << "Horsepower increased by " << hp << " HP" << endl;  }  int main() {  Car myCar(150);  Engine myEngine;  myCar.showHorsepower();  myEngine.increaseHorsepower(myCar, 50);  myCar.showHorsepower();  return 0;  } |

**OUTPUT:**



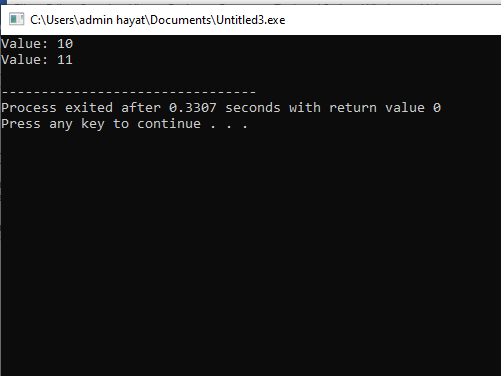
**Task 3**

Friend Function for Increment .

**CODE:**

|  |
| --- |
| #include <iostream>  using namespace std;  class Counter {  private:  int value;  public:  Counter(int val) : value(val) {}  void display() {  cout << "Value: " << value << endl;  }  friend void increment(Counter &c);  };  void increment(Counter &c) {  c.value++;  }  int main() {  Counter count(10);  count.display();  increment(count);  count.display();  return 0;  } |

**OUTPUT:**



**---------------------------------------------------**