**LAB 6: CLASSES AND OBJECTS MANIPULATION**

1. Consider the following **Progam6\_1.cpp and Program6\_2.cpp**.

[*Diberikan takrifan kelas* **acStaff** *di bawah.*]

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | **#include<iostream>**  **using namespace std;**  **class Test{**  **static int counter;**  **public:**  **void numberSoFar(){**  **counter++;**  **cout<<counter<<endl;**  **}**  **};**  **int Test::counter=0;**  **int main(){**  **Test a, b, c;**  **a.numberSoFar();**  **b.numberSoFar();**  **c.numberSoFar();**  **}** |

Program 6.1

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | **#include<iostream>**  **using namespace std;**  **class Test{**  **int counter;**  **public:**  **Test(){**  **counter=0;**  **}**  **void numberSoFar(){**  **counter++;**  **cout<<counter<<endl;**  **}**  **};**  **int main(){**  **Test a, b, c;**  **a.numberSoFar();**  **b.numberSoFar();**  **c.numberSoFar();**  **}** |

Program 6.2

1. Given the following program. Trace the program and determine the output.

*i.*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | **#include <iostream>**  **using namespace std;**  **class T{**  **static int i;**  **int j;**  **public:**  **T(int j){**  **this->j=j;**  **}**  **int getI(){**  **i++;**  **return i;**  **}**  **int getJ(){**  **j++;**  **return j;**  **}**  **};**  **int T::i=0;**  **int main(){**  **T t1(0), t2(0), t3(0);**  **cout<<"t1's static i="<<t1.getI()<<endl;**  **cout<<"and instance j="<<t1.getJ()<<endl;**  **cout<<"t2's static i="<<t2.getI()<<endl;**  **cout<<"t2's instance j="<<t2.getJ()<<endl;**  **cout<<"t3's static i="<<t3.getI()<<endl;**  **cout<<"t3's instance j="<<t3.getJ()<<endl;**  **}** |

Program 6.3

*ii.*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | **#include <iostream>**  **using namespace std;**  **class Test5{**  **public:**  **static int max(int, int);**  **};**  **int Test5::max(int x, int y){**  **int result;**  **if (x>y)**  **result=x;**  **else**  **result=y;**  **return result;**  **}**  **int main(){**  **int i=5;**  **int j=2;**  **int k=max(i, j);**  **cout<<"The max between\t" <<i<<"\tand\t"<<j<<"\tis\t"<<k;**  **}** |

Program 6.4

1. Consider the following **Accumulator** class definition.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | **class Accumulator**  **{**  **private:**  **int m\_nValue;**  **public:**  **Accumulator() { m\_nValue = 0; }**  **void Add(int nValue) { m\_nValue += nValue; }**  **// Make the Reset() function a friend of this class**  **friend void Reset(Accumulator &cAccumulator);**  **};** |

Program 6.5

1. Consider the following **Storage and Display** class definition.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | **class Storage**  **{**  **private:**  **int m\_nValue;**  **double m\_dValue;**  **public:**  **Storage(int nValue, double dValue)**  **{**  **m\_nValue = nValue;**  **m\_dValue = dValue;**  **}**  **// Make the Display class a friend of Storage**  **friend class Display;**  **};**  **class Display**  **{**  **private:**  **bool m\_bDisplayIntFirst;**  **public:**  **Display(bool bDisplayIntFirst) {**  **m\_bDisplayIntFirst = bDisplayIntFirst;**  **}**  **void DisplayItem(Storage &cStorage)**  **{**  **if (m\_bDisplayIntFirst)**  **cout << cStorage.m\_nValue << " " <<**  **cStorage.m\_dValue << std::endl;**  **else // display double first**  **cout << cStorage.m\_dValue << " " <<**  **cStorage.m\_nValue << std::endl;**  **}**  **};** |

Program 6.6

1. Discuss how **friend** allows a class to access to private members of the other class.
2. Answer the following questions on the use of memberwise assignment.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **p**  **class Date**  **{**  **int month, day, year;**  **public:**  **Date(int,int,int);**  **Date();**  **void print();**  **};** |

Program 6.7

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **class Date**  **{**  **int month, day, year;**  **public:**  **Date(int,int,int);**  **Date(const Date &second);**  **void print();**  **};** |

Program 6.8

1. The following code is intended to print out "6" but instead it prints out "5". How could we correct the error?

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **class Number**  **{**  **private:**  **int m\_num;**  **public:**  **Number(int n);**  **Number();**  **int GetNumber();**  **void SetNumber(int n);**  **};**  **Number::Number()**  **{**  **}**  **Number::Number(int n)**  **{**  **m\_num = n;**  **}**  **int Number::GetNumber()**  **{**  **return m\_num;**  **}**  **void Number::SetNumber(int n)**  **{**  **m\_num = n;**  **}**  **Number operator ++(Number &num)**  **{**  **return Number(num.GetNumber()+1);**  **}**  **int main()**  **{**  **Number num1(5);**  **++num1;**  **cout << num1.GetNumber() << endl;**  **return 0;**  **}** |

EXERCISE 2: STRUCTURED PROBLEMS

1. The following questions are based on the following class declaration in **Program6\_9.cpp**.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | **class Thing**  **{**  **private:**  **int x, y;**  **static int z;**  **public:**  **Thing()**  **{x=y=z;}**  **static void putThing(int a)**  **{**  **z = a;**  **}**  **};**  **int Thing::z=4;** |

Program 6.9

1. Given the following program with two classes.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  4041  42  43  44  45  46  47  48  49  50  51 | **#include <iostream>**  **#include <iomanip>**  **#include <cstring>**  **using namespace std;**  **class kawan2;**  **class kawan1**  **{**  **int value;**  **char girlF[10];**  **int age;**  **void secret() {strcpy(girlF, "Amy"); age = 30;}**  **public:**  **kawan1(int a) { value = a; secret();}**  **void print() { cout << " "<< value << " ";}**  **friend void tukarGanti(kawan1 &, kawan2 &);**  **};**  **class kawan2**  **{**  **int value;**  **int age;**  **char girlF[10];**  **void secret() {strcpy(girlF, "Lisa");age=22;}**  **public:**  **kawan2(int a) { value = a; secret();}**  **void print() { cout << " "<< value << " ";}**  **friend void tukarGanti(kawan1 &, kawan2 &);**  **};**  **void tukarGanti(kawan1 & x, kawan2 &y)**  **{**  **int temp = x.value;**  **x.value = y.value;**  **y.value = temp;**  **}**  **int main()**  **{**  **kawan1 objek1(100);**  **kawan2 objek2(50);**  **cout << "\nValue before function tukarGanti()";**  **objek1.print();**  **objek2.print();**  **tukarGanti(objek1,objek2);**  **cout << "\nValue after function tukarGanti()";**  **objek1.print();**  **objek2.print();**  **return 0;**  **}** |

Program 6.10

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  4041  42  43 | **#include <iostream>**  **#include <iomanip>**  **#include <cstring>**  **using namespace std;**  **class kawan2;**  **class kawan1**  **{**  **int value;**  **char girlF[10];**  **int age;**  **void secret() {strcpy(girlF, “Amy”); age = 30;}**  **public:**  **kawan1(int a) { value = a; secret();}**  **void print() { cout << “ “<< value << “ “;}**  **void gossip (kawan2);**  **friend class kawan2;**  **};**  **class kawan2**  **{**  **int value;**  **int age;**  **char girlF[10];**  **void secret() {strcpy(girlF, “Lisa”);age=22;}**  **public:**  **kawan2(int a) { value = a; secret();}**  **void print() { cout << “ “<< value << “ “;}**  **void gossip (kawan1);**  **friend void kawan1::gossip(kawan2);**  **};**  **// write the functions here**  **int main()**  **{**  **kawan1 objek1(100);**  **kawan2 objek2(50);**  **objek1.gossip(objek2);**  **objek2.gossip(objek1);**  **return 0;**  **}** |

Program 6.11

1. Given the following class definition of a **Rectangle**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **class Rectangle {**  **int width, height;**  **public:**  **Rectangle (int, int);**  **int calculateArea ();**  **friend Rectangle duplicate (Rectangle);**  **};** |

1. Implement the class with the following **main** function:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **int main () {**  **Rectangle rect(10,30), rectb;**  **rectb = duplicate (rect);**  **cout << rectb.calculateArea();**  **system("pause");**  **}** |

1. Now, change the Rectangle class to the following:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **class Square;**  **class Rectangle {**  **int width, height;**  **public:**  **Rectangle(int, int);**  **int calculateArea ();**  **void convert (Square);**  **};**  **class Square {**  **private:**  **int side;**  **public:**  **Square(int side){**  **this->side=side;**  **}**  **friend class Rectangle;**  **};** |

1. Consider the **FeetInches** class definition in Program 6\_13.cpp.

|  |  |
| --- | --- |
| **4**  **5**  **6**  **7**  **8**  **9**  **10**  **11**  **12**  **13**  **14**  **15**  **16**  **17**  **18**  **19**  **20**  **21**  **22**  **23** | **class FeetInches {**  **private:**  **int feet;**  **int inches;**  **public:**  **FeetInches(int f=0, int i=0){**  **feet = f;**  **inches = i;**  **}**  **void setFeet(int f)**  **{ feet = f; }**  **void setInches(int i)**  **{ inches = i;**  **}**  **int getFeet() const**  **{ return feet; }**  **int getInches() const**  **{ return inches; }**  **};** |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | **int main(){**  **FeetInches one(20,50);**  **FeetInches two(one);**  **FeetInches three;**  **three = one.multiply(two);**  **cout << one.getFeet() << endl**  **<< two.getFeet() << endl**  **<< three.getFeet() << endl;**  **}** |

1. Given the following class **Length**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | **class Length**  **{**  **private:**  **int len\_inches;**  **public:**  **Length (int feet, int inches)**  **{ len\_inches=12\*feet+inches;}**  **int getFeet(){ return len\_inches/12;}**  **int getInches(){return len\_inches%12;}**  **friend bool operator <(Length , Length );**  **friend bool operator ==(Length a, Length b);**  **};** |

Program 6.14

EXERCISE 3: PROBLEM SOLVING

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | **class Product**  **{**  **public:**  **Product( )**  **{ prodCode=0; price=0.0; }**  **int getProdCode( )**  **{ return prodCode; }**  **void setProdCode(int k)**  **{ prodCode=k; }**  **double getPrice( )**  **{ return price; }**  **void setPrice( double p)**  **{ price=p; }**  **friend ostream &operator<<( ostream&, const Product &);**  **private:**  **int prodCode;**  **double price;**  **};** |

Program 6.15