Metropolia
C++ Programming
Pasi Ranne

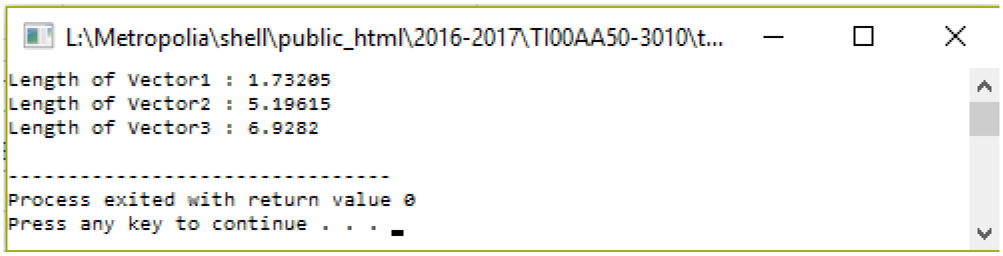
09/05/2017
Exercise 3 (class work)
Inheritance

Notice: Return your C++ source codes into Oma before deadline. Only so you can get credits from this exercise. You can't return these source codes after deadline.

1. **Binary operators overloading in C++**. Define class **Vector** which have double types attributes x, y and z. Implement a program where you overload binary operator **+**. Implement **getLength** which return the length of vector. Furthermore implement **setX**, **setY** and **setX** which sets the coordinates of vector (x,y,z). Further overload + operator to add two Vector objects. In main program define three Vector objects A, B and C. Further set coordinates to vectors A = (1, 1, 1) and B = (3, 3, 3). Then print lengths of vectors A and B. Further use + operator and add two objects A and B and set the result to C. At the end print length of vector C. Sample print is in figure 1.

   Reference: https://www.tutorialspoint.com/cplusplus/binary_operators_overloading.htm

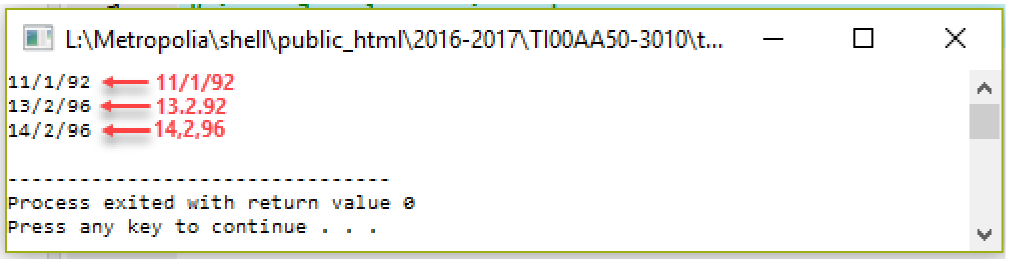
   Note! The right-hand operand is passed as an argument.


L:\Metropolia\shell\public_html\2016-2017\TI00AA50-3010\t...

```
Length of Vector1 : 1.73205
Length of Vector2 : 5.19615
Length of Vector3 : 6.9282

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Figure 1. Sample print in Dev C++ -program

2. **Constructor overloading**. Implement the class **dateT** in which constructor accepts date as a **string** in format **pp/k/vv** and **pp.kk.vv** or as three **integer** in order day, month and year. Implement method show, which shows the date in right format. Sample print is in figure 2. Red dates are parameters of the constructor.

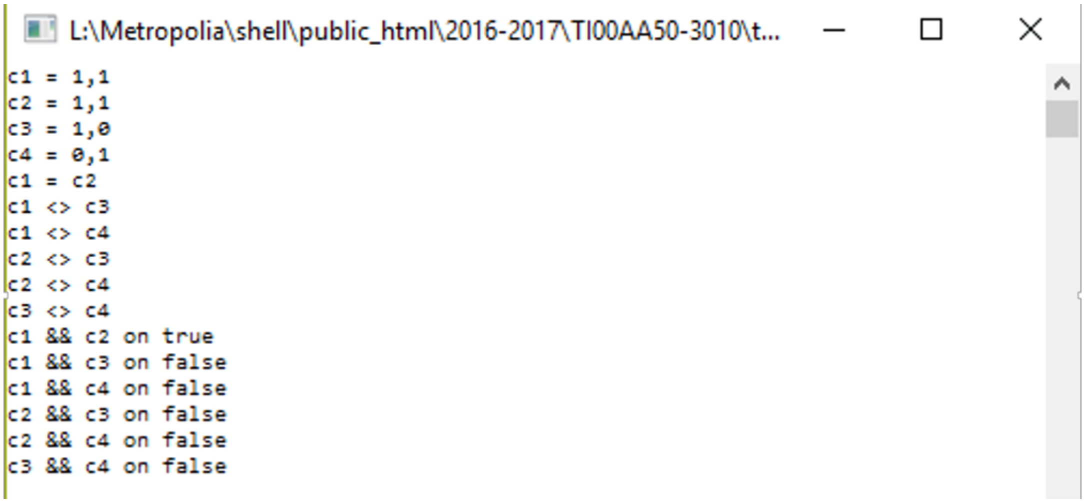

L:\Metropolia\shell\public_html\2016-2017\TI00AA50-3010\t...
11/1/92
13.2.92
14,2,96

```
11/1/92
13/2/96
14/2/96

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Figure 2. Sample print in Dev C++ -program

< continued >

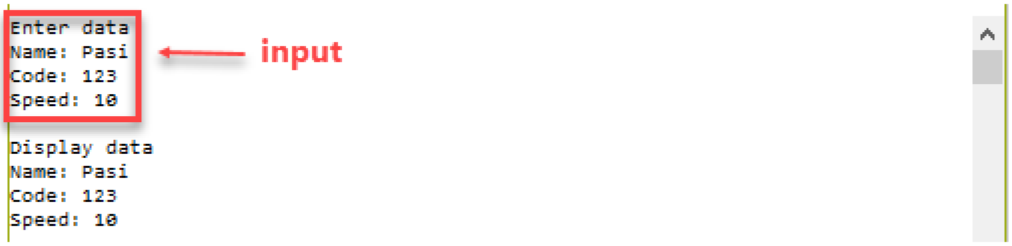
3. **Logical operators overloading**. Implement the class **coordinate** in which are two integer type attributes x and y. In this class is non parametric constructor and two parametric constructor. Furthermore in this class is method **get_xy**. Next you have to overload operators == and &&. In main function you have to create 4 objects with clause **coordinate c1(1, 1), c2(1, 1), c3(1, 0), c4(0, 1);**. After that you have to print coordinate values **c1**, **c2**, **c3** and **c4** (figure 6). You have to make comparisons if (c1 == c2), if (c1 == c3), if (c1 == c4), if (c2 == c3), if (c2 == c4), if (c3 == c4), if (c1 && c2), if (c1 && c3), if (c1 && c4), if (c2 && c3), if (c2 && c4) and if (c3 && c4). Sample print is in figure 3. Note! The right-hand operand is passed as an argument.


L:\Metropolia\shell\public_html\2016-2017\TI00AA50-3010\t...

```
c1 = 1,1
c2 = 1,1
c3 = 1,0
c4 = 0,1
c1 = c2
c1 <> c3
c1 <> c4
c2 <> c3
c2 <> c4
c3 <> c4
c1 && c2 on true
c1 && c3 on false
c1 && c4 on false
c2 && c3 on false
c2 && c4 on false
c3 && c4 on false
```

Figure 3. Sample print in Dev C++ -program

4. **Single Inheritance** (https://www.tutorialcup.com/cplusplus/inheritance.htm#single-inheritance). Implement a base class **Staff** with two private properties **char name[50]** and **int code**. Furthermore class **Staff** contains two public methods **getdata** and **display**. Implement a derived class **Typist** which is public. This class contains one private property **int speed** and two public methods **getdata** and **display**. In base class method **getdata** asks **name** and **code** (figure 4). In derived class method **getdata** asks **speed** (figure 4). In base class method **display** prints **name** and **code** (figure 4). In derived class method **display** prints **speed** (figure 4). In main function you must create one **Typist** object. Then you have to ask name, code and speed with **getdata** methods. Furthermore you have to print name, code and speed with **display** methods.

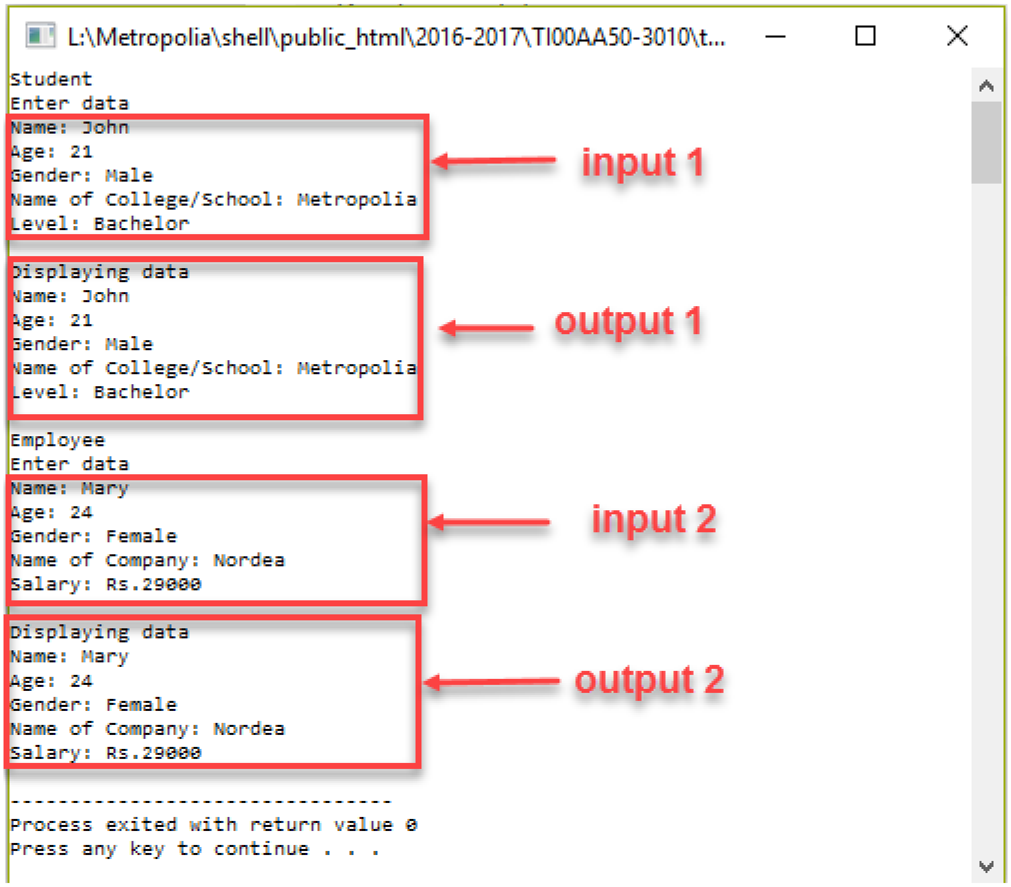


```
Enter data
Name: Pasi
Code: 123
Speed: 10

Display data
Name: Pasi
Code: 123
Speed: 10
```

Figure 4. Sample print in Dev C++ -program

< continued >

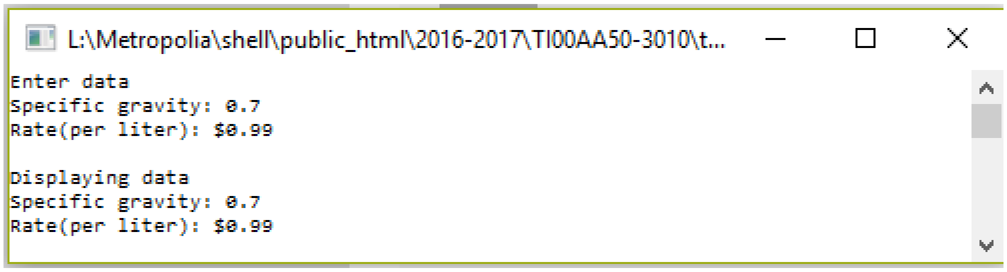
5. **Hierarchical inheritance** (https://www.tutorialcup.com/cplusplus/inheritance.htm#hierarchical-inheritance). Implement a base class **Person**, derived class **Student** and derived class **Employee**. The class **Person** contains properties **name**, **gender** and **age**. It contains also method **getdata** and **display**. Method **getdata** requests values of properties **name**, **gender** and **age** (figure 5, input 1 and input 2). Method **display** print values of properties **name**, **gender** and **age** (figure 5, output 1 and output 2). The class **Student** contains properties **institute** and **level**. It contains also method **getdata** and **display**. Method **getdata** requests values of properties **institute** and **level** (figure 5, input 1 and input 2). Method **display** print values of properties **institute** and **level** (figure 5, output 1 and output 2). The class **Employee** contains properties **company** and **salary**. It contains also method **getdata** and **display**. Method **getdata** requests values of properties **company** and **salary** (figure 5, input 1 and input 2). Method **display** print values of properties **company** and **salary** (figure 5, output 1 and output 2). In main program you have to create **Student** and **Employee** object. Then it prints text “Student” and text “Enter data” (figure 5). Furthermore you have to call method getdata of student and print text “Displaying data”. After that you have to call method display of student. Then it prints text “Empolyee” and text “Enter data” (figure 5). Furthermore you have to call method getdata of employee and print text “Displaying data”. After that you have to call method display of employee. Sample print is in figure 5.


L:\Metropolia\shell\public_html\2016-2017\TI00AA50-3010\t...
input 1
output 1
input 2
output 2

```
Student
Enter data
Name: John
Age: 21
Gender: Male
Name of College/School: Metropolia
Level: Bachelor

Displaying data
Name: John
Age: 21
Gender: Male
Name of College/School: Metropolia
Level: Bachelor

Employee
Enter data
Name: Mary
Age: 24
Gender: Female
Name of Company: Nordea
Salary: Rs.29000

Displaying data
Name: Mary
Age: 24
Gender: Female
Name of Company: Nordea
Salary: Rs.29000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Figure 5. Sample print in Dev C++ -program

6. **Multiple Inheritance**. Implement a class **Petrol** which inherits class **Fuel** and **Liquid**. In https://www.tutorialcup.com/cplusplus/inheritance.htm#multiple-inheritance you'll see the idea of **multiple Inheritance**. In class **Liquid** is one property **specific_gravity** and two methods. Input method request to give value of property **specific_gravity**. Output method prints a value of property **specific_gravity**. In class **Fuel** is one property **rate** and two methods. Input method request to give value of property **rate**. Output method prints a value of property **rate**. In class **Petrol** are two methods. Input method of class **Petrol** refers to input methods of both base classes. Furthermore Output method of class **Petrol** refers to output methods of both base classes. In main program you have to create one petrol object. Then you have to print text "Enter data" and refer to input method of **Petrol** class. In the end you have to print text "Displaying data" and refer to output method of **Petrol** class. Sample print is in figure 6.


L:\Metropolia\shell\public_html\2016-2017\TI00AA50-3010\t...

```
Enter data
Specific gravity: 0.7
Rate(per liter): $0.99

Displaying data
Specific gravity: 0.7
Rate(per liter): $0.99
```

Figure 6. Sample print in Dev C++ -program