VIETNAMESE-GERMAN UNIVSREITY
COMPUTER SCIENCE PROGRAM

Exam (Retake)
Course: Object Oriented Programming - C++
Duration: 120 minutes

Submit your code (.cpp files) to the classroom before 11:30 AM
You could use GG Search or eBooks.
Do not open any kinds of media, such as Facebook, Skype, etc., during the exam.

1) A library consists of a name (string) and a list of media (books and CDs). A book comprises a book name and a price. A CD has a CD ID and a price.

a) Implement classes for the above description. Implement the describe() method to print out the content of the objects

b) Write the addMedia(Media* media) method in the Library class to add a new media to the library. Create a library object that contains two books and one CD. Implement the describe() method to print out the library name and all information of media in the library.

```
int main() {
    // create a book with
    // name="C++", price=20.0
    book1.describe();

    // create a cd with
    // ID=12345, price=5.52
    cd1.describe();


    cout<<"\nLibrary Content" <<endl;
    // create a book with
    // name="C#", price=21.0
    // create a library with
    // name="Lib1"
    library.addMedia(&book1);
    library.addMedia(&book2);
    library.addMedia(&cd1);
    library.describe();

    return 0;
}
```

```
name: C++ ;price: 20
ID: 12345 ;price: 5.52
name: C# ;price: 21

Library Content
Lib Name: Lib1
name: C++ ;price: 20
name: C# ;price: 21
ID: 12345 ;price: 5.52
```

2) A book includes a name (string), an author (string) and publication date (that consists of day (int), month (int), and year (int).) The relationship between a book and a publication date in this problem can be a composition or generalized aggregation. This depends on customer requirements.

a) Implement the above description given that the relationship is determined to be composition. Create a book object to illustrate your implementation.

b) Implement the above description given that the relationship is determined to be generalized aggregation. Create a book object to illustrate your implementation.

```
int main() {
    Book book("C++", "Peter", 27, 6, 2022);
    book.describe();

    return 0;
}
```

```
Book name: C++ - Author name: Peter
Day: 27 - month: 6 - year: 2022
```

(a) Composition

```
int main() {
    Date pulicationDate(27, 6, 2022);
    Book book("C++", "Peter", &pulicationDate);
    book.describe();

    return 0;
}
```

```
Book name: C++ - Author name: Peter
Day: 27 - month: 6 - year: 2022
```

(b) generalized aggregation

3) Create a class template in which there are two attributes with T data type. Implement a sum method that allows computing summation between two numbers or two Circle. A circle includes a radius. The summation between two circles is the summation of their radius.

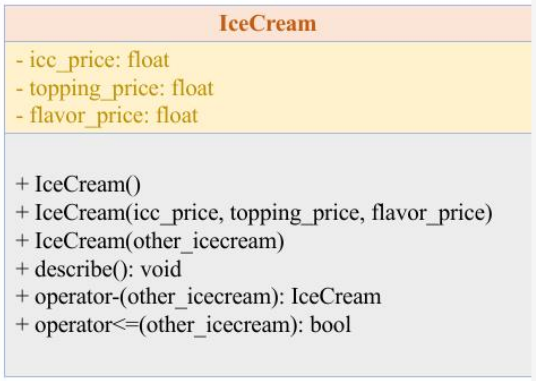
```
int main () {
   Data<int> intData(5, 6);
   cout << "sum 1: " << intData.sum() << endl;

   Circle circle1(5);
   Circle circle2(7);
   Data<Circle> circleData(circle1, circle2);
   Circle sumCircle = circleData.sum();
   sumCircle.describe();

   return 0;
}
```

```
sum 1: 11
radius: 12
```

4) Write a program to create a class called IceCream which is used to represent prices of an ice cream. The class will have three private float data members (ice price (ice cream cone), topping price, and flavor price). Add the following member methods:

a) a default constructor (set value of data members to zero)
b) a constructor with parameters
c) a copy constructor
d) overloaded subtraction operator (-), it must subtract ice price to ice price, topping price to topping price, and flavor price to flavor price
e) overloaded less than or equal operator (<=), two ice creams are considered to be less than or equal, if the three values of data members are less than or equal

Note: You have to implement the describe() method to print out the content of the objects (print values of ice price, topping price, and flavor price).




```cpp
int main() {
    // create IceCream ic0 with default constructor
    IceCream ic0;
    ic0.describe();

    // create IceCream ic1 with
    // icc_price=10.5 topping_price=2.3
    // flavor_price=1.2
    ic1.describe();

    // create IceCream ic2 with
    // icc_price=15 topping_price=5.2
    // flavor_price=7.5
    ic2.describe();

    // create IceCream ic3 by using copy constructor
    // copy ic1 into ic3
    ic3.describe();

    // overloaded subtraction operator (-)
    // ic3 = ic2 - ic1;
    ic3.describe();

    // overloaded less than or equal operator
    // is ic2 <= ic3
    cout << "Is ic2 <= ic3? " << (ic2 <= ic3) << endl;

    // assign ic1 to ic3
    // is ic3 <= ic1
    ic3 = ic1;
    ic3.describe();
    cout << "Is ic3 <= ic1? " << (ic3 <= ic1) << endl;
    return 0;
}
```

```
icc_price: 0 topping_price: 0 flavor_price: 0
icc_price: 10.5 topping_price: 2.3 flavor_price: 1.2
icc_price: 15 topping_price: 5.2 flavor_price: 7.5
icc_price: 10.5 topping_price: 2.3 flavor_price: 1.2
icc_price: 4.5 topping_price: 2.9 flavor_price: 6.3
Is ic2 <= ic3? 0
icc_price: 10.5 topping_price: 2.3 flavor_price: 1.2
Is ic3 <= ic1? 1
```