

# Object-oriented Programming

## Assignment Sheet No. 7

Date: December 6

### Exercise 7.1 (Custom Types: 3D-Points)

Implement a data type `point3D` for storing 3D-points. The data type shall have the following data members:

```
struct point3D {  
    int x, y, z;  
};
```

Add the following functionality to `point3D`:

- A default constructor and a constructor that takes three ints.
- A member function `void assign(int xc, int yc, int zc)` for assigning new values to the data members.
- Operators for testing for equality and inequality (`operator==` and `operator!=`).
- Input and output operators (`operator>>` and `operator<<`).
- An operator for adding to 3D-points (`operator+`).

Write a simple main program for testing your implementation of `point3D`.

### Exercise 7.2 (Custom Types: Time)

Consider the following data type for storing a timespan:

```
struct timespan {  
    int hours;    // >= 0  
    int minutes; // 0-59  
    int seconds; // 0-59  
};
```

Implement the following functionality for `time`:

- A member function `assign` which takes three integers as input and assigns these to the data members `hour`, `minute`, and `second`. Take care that only valid numbers are assigned.
- Write implementations of the following operators: `operator+`, `operator==`, and `operator<`.
- Implement input and output operators (`operator>>` and `operator<<`) for `time`. Choose an appropriate format for printing / reading a time.

Write a simple main program for testing your implementations.

### Exercise 7.3 (MP3 Collection)

We want to maintain a collection of MP3 records, which allows us to quickly look up all MP3 records for an artist.

- (i.) Implement a data type `MP3Record` for storing information about an MP3. The data type shall have the following data members:

```
struct MP3Record
{
    string  title;    // title of the song
    string  album;    // album containing the song
    timespan length; // length of the song
    int     year;     // year of publication
};
```

Use the `timespan` data type from Exercise 7.2 for storing the length of an MP3, implement a constructor for `MP3Record` that initializes all members, and implement an output operator for `MP3Record`.

- (ii.) Use a map that maps artists (of type `string`) to a vector of `MP3Records`, which stores all MP3s for that particular artist.

```
map<string, vector<mp3> > collection;
```

- (ii.) Write a main program that uses this map for maintaining a collection of MP3 records. The program shall start with an empty collection and allow the user to (a) insert new MP3 records and (b) lookup all MP3 records for an artist.