

## COEN 275 Assignment - 2

Q1. Implement a *User* class storing a username and number\_of\_followers (representing number of followers user have) as data members. Add a single constructor accepting two parameters, the username and number\_of\_followers. Provide appropriate getters (accessors) and setters (mutators). Add a copy constructor, move constructor and a destructor.

Answer:

a. Code ( adding a single 2-parameter constructor , getters and setters)

```
Assign2.cpp > User > getnumberof_followers()
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class User
7  {
8
9  private:
10     string username;
11     string number_of_followers;
12
13 public:
14     // Constructor accepting two parameters
15     User (string username, string followers)
16     {
17         username = username;
18         number_of_followers = followers;
19     }
20     // Getter for the username
21     string getuser_name()
22     {
23         return username;
24     }
25
26     // Setter for the username
27     void setuser_name(string username)
28     {
29         username = username;
30     }
31
32     // Getter for the number_of_followers
33     string getnumberof_followers()
34     {
35         return number_of_followers;
36     }
37
38     // Setter for the first name
39     void setnumberof_followers(string followers)
40     {
41         number_of_followers = followers;
42     }
43
44 };
45
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
rim@Rimas-MacBook-Air cpp test % cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
User : Rim 300k
rim@Rimas-MacBook-Air cpp test %
```

b. Code ( after adding a copy constructor, an assignment operator, and a destructor )

```
Assign2.cpp > main()
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class User
7  {
8
9  private:
10     string username;
11     string number_of_followers;
12
13 public:
14     // Constructor accepting two parameters
15     User (string usrname, string followers)
16     {
17         username = usrname;
18         number_of_followers = followers;
19     }
20
21     //Copy Constructor
22     User(const User &u1){
23         username = u1.username;
24         number_of_followers = u1.number_of_followers;
25     }
26     // Assignment Operator
27
28     User &operator=(const User &usr)
29     {
30         if(this!= &usr)
31         {
32             username = usr.username;
33             number_of_followers = usr.number_of_followers;
34         }
35         return *this;
36     }
37
38     // Destructor
39
40     ~User()
41     {}
42 }
```

```

Assign2.cpp > main()
43 // Getter for the username
44 string getuser_name()
45 {
46     return username;
47 }
48
49 // Setter for the username
50 void setuser_name(string username)
51 {
52     username = username;
53 }
54
55 // Getter for the number_of_followers
56 string getnumberof_followers()
57 {
58     return number_of_followers;
59 }
60
61 // Setter for the first name
62 void setnumberof_followers(string followers)
63 {
64     number_of_followers = followers;
65 }
66
67 };
68
69
70
71 int main()
72 {
73     User u1("Rim", "300k"); //Normal constructor is called here
74     User u2 = u1; // Copy constructor is called here
75
76     // Access values assigned by constructors
77     cout << "User1 : " << u1.getuser_name() << " , Followers1 = " << u1.getnumberof_followers() << endl;
78     cout << "User2 : " << u2.getuser_name() << " , Followers2 = " << u2.getnumberof_followers() << endl;
79
80     // Test assignment operator
81     User u3("Jerry", "3.5M");
82     u1 = u3;
83     cout << "After assignment :\n" "User1 = " << u1.getuser_name() << " , Followers1 = " << u1.getnumberof_followers() << endl;
84
85     return 0;
86 }

```

Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
rim@Rimas-MacBook-Air cpp test % cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
User1 : Rim , Followers1 = 300k
User2 : Rim , Followers2 = 300k
After assignment :
User1 = Jerry , Followers1 = 3.5M
rim@Rimas-MacBook-Air cpp test %

```

Q2. Add a new data member to store the number\_of\_following (representing number of users a user follow) of a user and provide a getter & setter. Add a new constructor that accepts three parameters, a username, number\_of\_followers and number\_of\_following. Modify the original two parameter constructor to automatically set number\_of\_following for a given username and number\_of\_followers and delegate the actual construction work to the new three parameter constructor.

```
Assign2.cpp > User
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class User
7  {
8
9  private:
10     string username;
11     string number_of_followers;
12     string number_of_following;
13
14 public:
15     // Constructor accepting two parameters
16     User (string usname, string followers, string no_of_following)
17     {
18         username = usname;
19         number_of_followers = followers;
20         number_of_following = no_of_following;
21     }
22
23     // Getter for the username
24     string getuser_name()
25     {
26         return username;
27     }
28
29     // Setter for the username
30     void setuser_name(string usname)
31     {
32         username = usname;
33     }
34
35     // Getter for the number_of_followers
36     string getnumberof_followers()
37     {
38         return number_of_followers;
39     }
40
41     // Setter for the first name
42     void setnumberof_followers(string followers)
43     {
44         number_of_followers = followers;
45     }
46 }
```

```
Assign2.cpp > User
46
47 // Getter for the number_of_following
48 string getnumberof_following()
49 {
50     return number_of_following;
51 }
52
53 // Setter for the number_of_following
54 void setnumberof_following(string no_of_following)
55 {
56     number_of_following = no_of_following;
57 }
58
59 };
60
61
62 int main()
63 {
64     User u("Rim","388k", "1k");
65
66     cout << "User : " << u.getuser_name() << " , Followers = " << u.getnumberof_followers() << " , Following = " << u.getnumberof_following() << endl;
67
68     return 0;
69 }
70 }
```

Output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
● rim@Rimas-MacBook-Air cpp test % cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
User : Rim , Followers = 300k , Following = 1k
○ rim@Rimas-MacBook-Air cpp test %
```

Q3. Take the *user* class from Q1, and add a derived class called *Creator*. The creator class adds two data members, a *creator\_id* and *number\_of\_reels\_created*. Provide appropriate constructor. From *Creator*, derive two more classes called *TechCreator* and *NonTechCreator*.

```
Assign2.cpp > main()
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class User
7  {
8
9  private:
10     string username;
11     string number_of_followers;
12     string number_of_following;
13
14 public:
15     // Constructor accepting two parameters
16     User (string usrname, string followers, string no_of_following)
17     {
18         username = usrname;
19         number_of_followers = followers;
20         number_of_following = no_of_following;
21     }
22
23     // Getter for the username
24     string getuser_name()
25     {
26         return username;
27     }
28
29     // Setter for the username
30     void setuser_name(string usrname)
31     {
32         username = usrname;
33     }
34
35     // Getter for the number_of_followers
36     string getnumberof_followers()
37     {
38         return number_of_followers;
39     }
40
41     // Setter for the number_of_followers
42     void setnumberof_followers(string followers)
43     {
44         number_of_followers = followers;
45     }
46
```

```

G: Assign2.cpp > main()
47 // Getter for the number_of_following
48 string getnumberof_following()
49 {
50     return number_of_following;
51 }
52
53 // Setter for the number_of_following
54 void setnumberof_following(string no_of_following)
55 {
56     number_of_following = no_of_following;
57 }
58
59 };
60
61 class Creator : public User
62 {
63 private:
64     int creator_id;
65     int number_of_reels_created;
66
67 public:
68 // Constructor for Creator class
69 Creator (int crt_id,string username, string followers, string no_of_following, int no_of_reels_created) : User (username, followers, no_of_following)
70 {
71     creator_id = crt_id;
72     number_of_reels_created = no_of_reels_created;
73 }
74 // Getter for the creator_id
75 int getid()
76 {
77     return creator_id;
78 }
79
80 // Setter for the creator_id
81 void setid(int crt_id)
82 {
83     creator_id = crt_id;
84 }
85
86 // Getter for the number_of_reels_created
87 int getnumb_reels_created()
88 {
89     return number_of_reels_created;
90 }
91
92

```

```

// Setter for the number_of_reels_created
void setnumb_reels_created(int no_of_reels_created)
{
    number_of_reels_created = no_of_reels_created;
}

};

class TechCreator : public Creator
{
private:
    int creator_id;
    int number_of_reels_created;

public:
    TechCreator (int crt_id,string username, string followers, string no_of_following, int no_of_reels_created) : Creator (crt_id, username, followers, no_of_following)
    {
        creator_id = crt_id;
        number_of_reels_created = no_of_reels_created;
    }
};

class NonTechCreator : public Creator
{
private:
    int creator_id;
    int number_of_reels_created;

public:
    NonTechCreator (int crt_id,string username, string followers, string no_of_following, int no_of_reels_created) : Creator (crt_id, username, followers, no_of_following)
    {
        creator_id = crt_id;
        number_of_reels_created = no_of_reels_created;
    }
};

```

```

int main()
{
    Creator c(101, "Rim", "300k", "1k", 60);
    TechCreator tc(102, "Jerry", "30k", "2k", 50);
    NonTechCreator nontc(103, "Tim", "3k", "10k", 30);

    cout << "User ID = " << c.getid() << " ,User : " << c.getuser_name() << " , Followers = "
    << c.getnumberof_followers() << " , Following = " << c.getnumberof_following() << " ,Number of Reels created = "
    << c.getnumb_reels_created() << endl;

    cout << "User ID = " << tc.getid() << " ,User : " << tc.getuser_name() << " , Followers = "
    << tc.getnumberof_followers() << " , Following = " << tc.getnumberof_following() << " ,Number of Reels created = "
    << tc.getnumb_reels_created() << endl;

    cout << "User ID = " << nontc.getid() << " ,User : " << nontc.getuser_name() << " , Followers = "
    << nontc.getnumberof_followers() << " , Following = " << nontc.getnumberof_following() << " ,Number of Reels created = "
    << nontc.getnumb_reels_created() << endl;

    return 0;
}

```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
rim@Rimas-MacBook-Air cpp test % cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
User ID = 101 ,User : Rim , Followers = 300k , Following = 1k ,Number of Reels created = 60
User ID = 102 ,User : Jerry , Followers = 30k , Following = 2k ,Number of Reels created = 50
User ID = 103 ,User : Tim , Followers = 3k , Following = 10k ,Number of Reels created = 30
rim@Rimas-MacBook-Air cpp test %
```

Q4. Continuing with your solution, add *introduceMe()* method to the *User* class returning a string representation of a user. Override this method in the *Creator*, *TechCreator* and *NonTechCreator* classes to build up a complete string representation by delegating part of their work to parent classes.

```
G: Assign2.cpp > main()
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class User
7  {
8
9  private:
10     string username;
11     string number_of_followers;
12     string number_of_following;
13
14 public:
15     // Constructor accepting two parameters
16     User (string usrname, string followers, string no_of_following)
17     {
18         username = usrname;
19         number_of_followers = followers;
20         number_of_following = no_of_following;
21     }
22
23     // Getter for the username
24     string getuser_name()
25     {
26         return username;
27     }
28
29     // Setter for the username
30     void setuser_name(string usrname)
31     {
32         username = usrname;
33     }
34
35     // Getter for the number_of_followers
36     string getnumberof_followers()
37     {
38         return number_of_followers;
39     }
40
41     // Setter for the number_of_followers
42     void setnumberof_followers(string followers)
43     {
44         number_of_followers = followers;
45     }
46
```

```

G: Assign2.cpp > User
46
47 // Getter for the number_of_following
48 string getnumberof_following()
49 {
50     return number_of_following;
51 }
52
53 // Setter for the number_of_following
54 void setnumberof_following(string no_of_following)
55 {
56     number_of_following = no_of_following;
57 }
58
59 virtual string introduceMe()
60 {
61     return
62     "MyName = " + getuser_name() + " " +
63     "Followers = " + getnumberof_followers() + " " +
64     "Following = " + getnumberof_following();
65 }
66 };
67
68 class Creator : public User
69 {
70 private:
71     int creator_id;
72     int number_of_reels_created;
73
74 public:
75     // Constructor for Creator class
76     Creator (int crt_id,string username, string followers, string no_of_following, int no_of_reels_created) : User (username, followers, no_of_following)
77     {
78         creator_id = crt_id;
79         number_of_reels_created = no_of_reels_created;
80     }
81     // Getter for the creator_id
82     int getid()
83     {
84         return creator_id;
85     }
86
87     // Setter for the creator_id
88     void setid(int crt_id)
89     {
90         creator_id = crt_id;
91     }

```

```

G: Assign2.cpp > NonTechCreator > NonTechCreator(int, string, string, string, int)
93 // Getter for the number_of_reels_created
94 int getnumb_reels_created()
95 {
96     return number_of_reels_created;
97 }
98
99 // Setter for the number_of_reels_created
100 void setnumb_reels_created(int no_of_reels_created)
101 {
102     number_of_reels_created = no_of_reels_created;
103 }
104
105 string introduceMe()
106 {
107     return "MyID = " + to_string(getid()) + " " + Creator :: introduceMe() + "My Total Reels = " + to_string(getnumb_reels_created());
108 }
109 };
110
111 class TechCreator : public Creator
112 {
113 private:
114     int creator_id;
115     int number_of_reels_created;
116
117 public:
118     TechCreator (int crt_id,string username, string followers, string no_of_following, int no_of_reels_created) :
119     Creator (crt_id, username, followers, no_of_following, no_of_reels_created)
120     {
121         creator_id = crt_id;
122         number_of_reels_created = no_of_reels_created;
123     }
124     // Overriding the method introduceMe()
125     string introduceMe()
126     {
127         return "TechCreator : " + Creator::introduceMe();
128     }
129 };

```



```

Assign2.cpp > NonTechCreator > NonTechCreator(int, string, string, string, int)
130
131 class NonTechCreator : public Creator
132 {
133     private:
134         int creator_id;
135         int number_of_reels_created;
136
137     public:
138         NonTechCreator (int crt_id,string username, string followers, string no_of_following, int no_of_reels_created)
139         {
140             Creator (crt_id, username, followers, no_of_following, no_of_reels_created)
141         }
142         {
143             creator_id = crt_id;
144             number_of_reels_created = no_of_reels_created;
145         }
146         // Overriding the method introduceMe()
147         string introduceMe()
148         {
149             return "NonTechCreator : " + Creator::introduceMe();
150         }
151     };
152
153 int main()
154 {
155     Creator c(101, "Rim","300k", "1k", 60);
156     TechCreator tc(102, "Jerry","30k", "2k", 50);
157     NonTechCreator nontc(103, "Tim","3k", "10k", 30);
158
159     cout << "User ID = " << c.getid() << " ,User : " << c.getuser_name() << " , Followers = "
160     << c.getnumberof_followers() << " , Following = " << c.getnumberof_following() << " ,Number of Reels created = "
161     << c.getnumb_reels_created() << " \nIntroduce - " << c.introduceMe()<< endl;
162
163     cout << "User ID = " << tc.getid() << " ,User : " << tc.getuser_name() << " , Followers = "
164     << tc.getnumberof_followers() << " , Following = " << tc.getnumberof_following() << " ,Number of Reels created = "
165     << tc.getnumb_reels_created() << " \nIntroduce - " << tc.introduceMe()<< endl;
166
167     cout << "User ID = " << nontc.getid() << " ,User : " << nontc.getuser_name() << " , Followers = "
168     << nontc.getnumberof_followers() << " , Following = " << nontc.getnumberof_following() << " ,Number of Reels created = "
169     << nontc.getnumb_reels_created() << " \nIntroduce - " << nontc.introduceMe()<< endl;
170     return 0;
171 }

```

Output:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
rim@Rimas-MacBook-Air cpp test % cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
User ID = 101 ,User : Rim , Followers = 300k , Following = 1k ,Number of Reels created = 60
zsh: segmentation fault  "/Users/rim/cpp test/"tempCodeRunnerFile
rim@Rimas-MacBook-Air cpp test % cd "/Users/rim/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rim/cpp test/"tempCodeRunnerFile
User ID = 101 ,User : Rim , Followers = 300k , Following = 1k ,Number of Reels created = 60
zsh: segmentation fault  "/Users/rim/cpp test/"tempCodeRunnerFile
rim@Rimas-MacBook-Air cpp test %

```

Q5. Define a vector to store a mix of Creator, TechCreator and NonTechCreator and fill it with some test data. Finally use a range based for loop to call *introduceMe()* on all of the elements in the vector.

```
Assign2.cpp ×
Assign2.cpp > Creator > Creator(int, string, string, string, int)
1  #include <iostream>
2  #include <string>
3  #include <vector>
4
5  using namespace std;
6
7  class User
8  {
9
10 private:
11     string username;
12     string number_of_followers;
13     string number_of_following;
14
15 public:
16     // Constructor accepting two parameters
17     User (string usrname, string followers, string no_of_following)
18     {
19         username = usrname;
20         number_of_followers = followers;
21         number_of_following = no_of_following;
22     }
23
24     // Getter for the username
25     string getuser_name()
26     {
27         return username;
28     }
29
30     // Setter for the username
31     void setuser_name(string usrname)
32     {
33         username = usrname;
34     }
35
36     // Getter for the number_of_followers
37     string getnumberof_followers()
38     {
39         return number_of_followers;
40     }
41
42     // Setter for the number_of_followers
43     void setnumberof_followers(string followers)
44     {
45         number_of_followers = followers;
46     }
47 }
```

Assign2.cpp X

Assign2.cpp > Creator > Creator(int, string, string, string, int)

```
48 // Getter for the number_of_following
49 string getnumberof_following()
50 {
51     return number_of_following;
52 }
53
54 // Setter for the number_of_following
55 void setnumberof_following(string no_of_following)
56 {
57     number_of_following = no_of_following;
58 }
59
60 virtual string introduceMe()
61 {
62     return
63         "MyName = " + getuser_name() + " " +
64         "Followers = " + getnumberof_followers() + " " +
65         "Following = " + getnumberof_following();
66 }
67 };
68
69 class Creator : public User
70 {
71     private:
72         int creator_id;
73         int number_of_reels_created;
74
75     public:
76         // Constructor for Creator class
77         Creator (int crt_id, string username, string followers, string no_of_following, int no_of_reels_created) : User (username, followers, no_of_following)
78         {
79             creator_id = crt_id;
80             number_of_reels_created = no_of_reels_created;
81         }
82         // Getter for the creator_id
83         int getid()
84         {
85             return creator_id;
86         }
87
88         // Setter for the creator_id
89         void setid(int crt_id)
90         {
91             creator_id = crt_id;
92         }
93 }
```

Assign2.cpp X

Assign2.cpp > Creator > Creator(int, string, string, string, int)

```
94 // Getter for the number_of_reels_created
95 int getnumb_reels_created()
96 {
97     return number_of_reels_created;
98 }
99
100 // Setter for the cnumber_of_reels_created
101 void setnumb_reels_created(int no_of_reels_created)
102 {
103     number_of_reels_created = no_of_reels_created;
104 }
105
106 string introduceMe()
107 {
108     return "MyID = " + to_string(getid()) + " " + Creator :: introduceMe() + "My Total Reels = " + to_string(getnumb_reels_created());
109 }
110 };
111
112 class TechCreator : public Creator
113 {
114     private:
115         int creator_id;
116         int number_of_reels_created;
117     public:
118         TechCreator (int crt_id,string usrxname, string followers, string no_of_following, int no_of_reels_created) :
119             Creator (crt_id, usrxname, followers, no_of_following, no_of_reels_created)
120         {
121             creator_id = crt_id;
122             number_of_reels_created = no_of_reels_created;
123         }
124         // Overriding the method introduceMe()
125         string introduceMe()
126         {
127             return "TechCreator : " + Creator::introduceMe();
128         }
129     };
130 };
131
```

```

Assign2.cpp > Creator > Creator(int, string, string, string, int)
131
132 class NonTechCreator : public Creator
133 {
134     private:
135         int creator_id;
136         int number_of_reels_created;
137
138     public:
139         NonTechCreator (int crt_id, string username, string followers, string no_of_following, int no_of_reels_created)
140         : Creator (crt_id, username, followers, no_of_following, no_of_reels_created)
141         {
142             creator_id = crt_id;
143             number_of_reels_created = no_of_reels_created;
144         }
145         // Overriding the method introduceMe()
146         string introduceMe()
147         {
148             return "NonTechCreator : " + Creator::introduceMe();
149         }
150     };
151
152
153 int main()
154 {
155     vector<User> users;
156     users.push_back(User( "Rim", "5.6M", "1k"));
157     users.push_back(User( "Tim", "1M", "20k"));
158     users.push_back(User( "Jim", "2M", "1k"));
159
160     for (auto user : users)
161     {
162         cout << user.introduceMe() << endl;
163     }
164
165     return 0;
166 }

```

Output:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
cd "/Users/rin/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rin/cpp test/"tempCodeRunnerFile
rim@Rinas-MacBook-Air cpp test % cd "/Users/rin/cpp test/" && g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile && "/Users/rin/cpp test/"tempCodeRunnerFile
tempCodeRunnerFile.cpp:160:10: warning: 'auto' type specifier is a C++11 extension [-Wc++11-extensions]
    for (auto user : users)
        ^
tempCodeRunnerFile.cpp:160:20: warning: range-based for loop is a C++11 extension [-Wc++11-extensions]
    for (auto user : users)
        ^
2 warnings generated.
MyName = Rim Followers = 5.6M Following = 1k
MyName = Tim Followers = 1M Following = 20k
MyName = Jim Followers = 2M Following = 1k
rim@Rinas-MacBook-Air cpp test %

```

