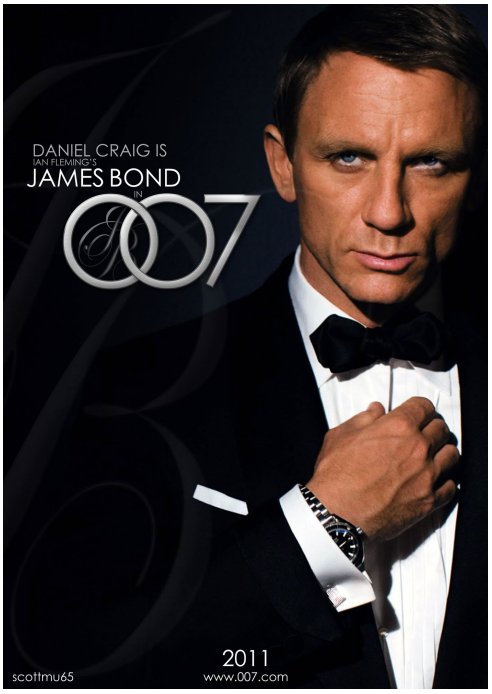
Military Institute of Science & Technology
Department of Computer Science and Engineering

CSE 206: Object Oriented Programming Language Sessional
Level – 2, Spring Term, Online 01
Date – 06 May 2021

"The name's Bond.... James Bond".



Who hasn't heard this iconic quote from Ian Fleming's fictional series **James Bond?** You will have the pleasure of being a part of his mission today.

The British Secret Service agent is chasing after his archenemy **Ernst Stavro Blofeld** in Portimão, the southern coast of Portugal. **Agent 007** retrieved a piece of harddrive from Blofeld's office last night. Today in the morning, he's meeting his **Quartermaster** in a museum where he asks for his help to find what's inside the drive. **Q** discovered that there's a code written in C++ and a documentation of the classes used in the code. Mr Bond asked Q to replace the classes and he'll use this drive as a bait to catch Blofeld.

But before that Q needs to find out the definition of the classes. Here's where you come in. Given the code snippet found in the hard drive and the documentation of the class files, you have to write the definition of the classes for Q. But there's not much time on the clock. You only have **75 mins.**

Bond and Q plotting against Blofeld in the museum.

Code Snippet

```
int main(){
	Database db;
	db.addUser(User("Eva", "green"));
	db.addUser(User("Halle", "beryy"));
	db.addUser(User("Jane", "seymour"));
	db.addUser(User("Rosamund", "pike"));
	db.addUser(User("Monica", "belucci"));
	db.print();
	db.addUser(User("Gemma", "arterton"));
	db.addUser(User("Lea", "seydoux"));

	User test_user1("Rosamund", "pike");
	User test_user2("Rosamund", "berry");
	User test_user3("Samantha", "pike");

	if(login(db, test_user1))
		cout << "Login Successful\n";
	else
		cout << "Login Failed\n";

	if(login(db, test_user2))
		cout << "Login Successful\n";
	else
		cout << "Login Failed\n";

	if(login(db, test_user3))
		cout << "Login Successful\n";
	else
		cout << "Login Failed\n";

	return 0;
}
```

Documentation of the Classes

```
// include header files
// forward declaration if required


class User {
      // name of user as character pointer
      // hash of password as long long
public:
      // Constructors, Destructors and Copy Constructors as necessary

      // In the constructor convert name to lowercase letters and concat
      // name after password (password will always be lowercase). Then
      // call hash_value function to assign the value of hash variable

      long long hash_value(string pass){
            const int p = nearest prime number to your student ID % 100
      /*

         Hash value calculation from pass
         if pass is "xyz"
         hash = (24 * p * 1) + (25 * p * 2) + (26 * p * 3)
              = summation_of(letter_position_in_alphabet * p *
                                                    string_position)
         Note: Don't use the ASCII values.
      */
      }

      // Use friend function / class as necessary

};

class Database{
      User * users;
      int cur_user;
      int max_user;
public:
      // Constructors, Destructors and Copy Constructors as necessary

      // Whenever a Database class object is created users array will be
      // initialized with length 4. One User named "Admin" with password
      // "admin" will be created and stored in the first position of the
      // users array.
```

```
      void addUser(User ob){
            if(cur_user < max_user){
                  // add new user in the next available position
            }
            else {
                  // double the size of max_user.
                  // dynamically create a tmp User array of size
                  // max_user.

                   memcpy(tmp, users, sizeof(User) * max_user);
                  // this will copy the contents of users to tmp

                  // delete old users array
                  // assign tmp to users (users = tmp)
                  // Now add new user in the next available position
            }
            // Don't forget to increment cur_user
      }

      void print(){
            // Will print the current user count and user names in db.
            // See output for formatting
      }
      // Use friend function / class as necessary
};

bool login(Database d, User a){
      // Iterate through the users in d and if username and hash matches
      // the username and hash of a, return true.
      // Else return false.

}
```

Output

```
Current User No: 6
Current User's List:
1 Admin
2 Eva
3 Halle
4 Jane
5 Rosamund
6 Monica
Login Successful
Login Failed
Login Failed
```

Marks Distribution

Criteria	**Marks**
Structure of class	4
Constructor, Destructor & Copy Constructor	13
Member and non-member function definition	11
Overall completeness	2
Total	**30**