code.sprint MT

TASK BOOKLET

- Qualifiers Round -

Post-Secondary Category

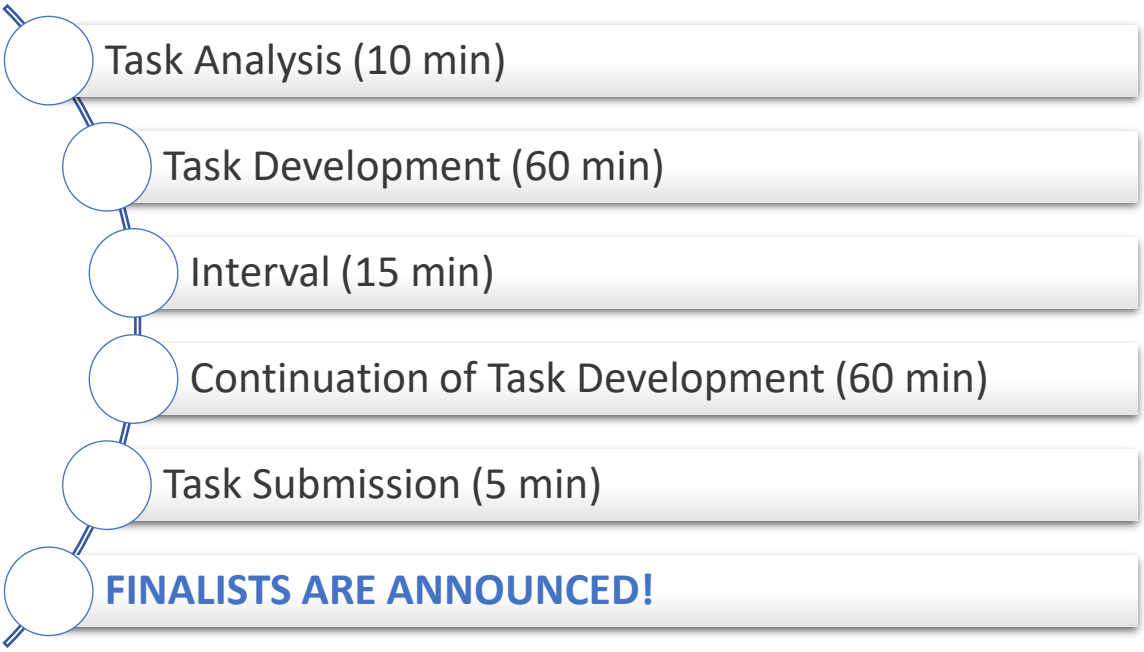
2021

DIRECTORATE FOR LEARNING &
ASSESSMENT PROGRAMMES

x

ICE Malta

Qualifiers Round Schedule



Password Encryption (120 minutes)

Let's say you set up an account at TheMostSecureWebsite.com. You type in your username and password and set up your account. A little while later you receive an email informing you that, ironically, the website has been hacked, and the usernames and passwords of every user, which were stored in *plaintext*, are now for sale on the dark web. You panic! You start changing the passwords on all your accounts. But wait, you used the same password for all your accounts now - you moron! You wonder, "Isn't that a bad idea? Shouldn't my password be stored as ciphertext (secret code) so hackers can't just read it?"

You are correct; any web app or service that uses a username/password login system should be storing their users' passwords using some form of encryption method.

Create a login system that allows the user to login or register a new account. Your system should simulate both the client, the user trying to login/register, and the server that stores the users' credentials.

Functionality #1: Main Menu

1. The system presents a menu with three options: 1) Login, 2) Register, 3) Exit.
    - **Login:** User must insert the username and password to have access to the system. If login credentials are correct the program displays an access granted message and the program stops; otherwise the main menu keeps on looping.
    - **Register:** User will register a new account by inserting the username and password. When the user registers, the system generates an encryption key and the server stores the username, the encryption key, and the encrypted password.
    - **Exit:** Quits the Main Menu and the program stops.

Functionality #2: Validation

1. The user's menu option must be validated; non-existing options or invalid input must be handled accordingly.
2. Username must be unique.
3. Username should not include spaces and have a minimum of four characters.
4. Passwords should have a minimum of eight characters and contain at least one:
   - lowercase and uppercase letter,
   - number, and
   - special character from these: '!', '@', '#', '$', '%', '&', '*'

   * Therefore, the complete set of characters that a password can contain is as follows:

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
x	y	z	A	B	C	D	E	F	G	H	I	J	K	L	M	O	P	Q	R	S	T	U
V	W	X	Y	Z	0	1	2	3	4	5	6	7	8	9	!	@	#	$	%	&	*	

5. Encrypted keys cannot have duplicate characters.

Functionality #3: Log In, Registering and Cryptography

1. The system uses a **Substitution Cipher** algorithm that randomly substitutes the characters used in the password by other characters. This is achieved by generating a random encryption key in this format: lowercase letter, number, uppercase letter, special character, lowercase letter, number, uppercase letter, special character. For example: g3W*a0T!

   The sequence of the set of available characters will then start with the generated encrypted key and continues with the remaining characters in the set. Check the encrypted set of characters below if the key **g3W*a0T!** is used.

Original Set:	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
	x	y	z	A	B	C	D	E	F	G	H	I	J	K	L	M	O	P	Q	R	S	T	U
	V	W	X	Y	Z	0	1	2	3	4	5	6	7	8	9	!	@	#	$	%	&	*	

Encrypted Set:	**g**	**3**	**W**	*****	**a**	**0**	**T**	**!**	b	c	d	e	f	h	i	j	k	l	m	n	o	p	q
	r	s	t	u	v	w	x	y	z	A	B	C	D	E	F	G	H	I	J	K	L	M	O
	P	Q	R	S	U	V	X	Y	Z	1	2	4	5	6	7	8	9	@	#	$	%	&	

   If the password entered is **Hello$123**, its encrypted version is **Baeei#XYZ**

2. When the user registers a new account, the system generates a random encryption key which is transferred to the server along the username and password (plaintext). From the server's side, the username, encrypted key and encrypted password (ciphertext) are stored.

3. When the user logs in, the server checks for the username and if it is available, the server encrypts the submitted password using the corresponding username's encryption key.

Functionality #4: Data Initialization

The following users are to be registered and stored in the server:

Username	Password
johnBorg	Joe!King
mariaCalleja	*callejaMarie*3

Name the class containing the main method **RunApp.**

Submit your program in a folder named **Encrypt_Pass**

Functionality #5: Displaying status messages

1. When the user registers a new account, the system should display the data that the server stores; i.e. username, random generated encryption key, and the encrypted password. *Check Screenshot 1 below.*
2. When the user logs in, the system should display the encrypted password (if the username exists), and whether the user has been logged in successfully or not. *Check Screenshot 2 below.*

```
******* REGISTER ACCOUNT *******

>> Username: jhonnyBest
>> Password: King$1979

-------------------------------------
|  Account registered successfully  |
-------------------------------------

Details saved on server:
>> Username: jhonnyBest
>> Encryption Key: q9G&l4R*
>> Encrypted Password: EafR@X636

...press Enter to continue
```

Screenshot 1: Register New Account

```
******* LOG IN *******

>> Username: jbor
>> Password: Hello$123

Encrypted Password: Baeei#XYZ

-----------------------------
|  Logged In Successfully   |
-----------------------------
...press Enter to continue
```

Screenshot 2: Log In

Assessment Rubric

Program Functionality	User-Friendly Interface	Proper use of Comments	Proper Conventions (Camel case, meaningful var names etc.)	Name of Folder & Class/es	User Input	Suitable Prompts / Messages displayed
Generating Encryption Key	Generating Encrypted Password	Proper use of data structures and/or files	Modular Code	Data Initialisation	Code Efficiency	Use of GUI (Extra Feature)
Validation				Other Features (Not listed in the task)	**Maximum Score: 38** **+ 2 for every extra feature.**	
Menu Option	Username	Password	Encryption Key			
0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily						


eSkills
eSkills Malta Foundation


HSBC | Malta Foundation


melita foundation


Tech.mt
MALTA - LEADING THROUGH INNOVATION


FreeHour