Fundamentals of Structured Programming

Lecture 6

Functions I (Built-In)

Course Coordinator: Prof. Zaki Taha Fayed

Presented by: Dr. Sally Saad

SallySaad@gmail.com

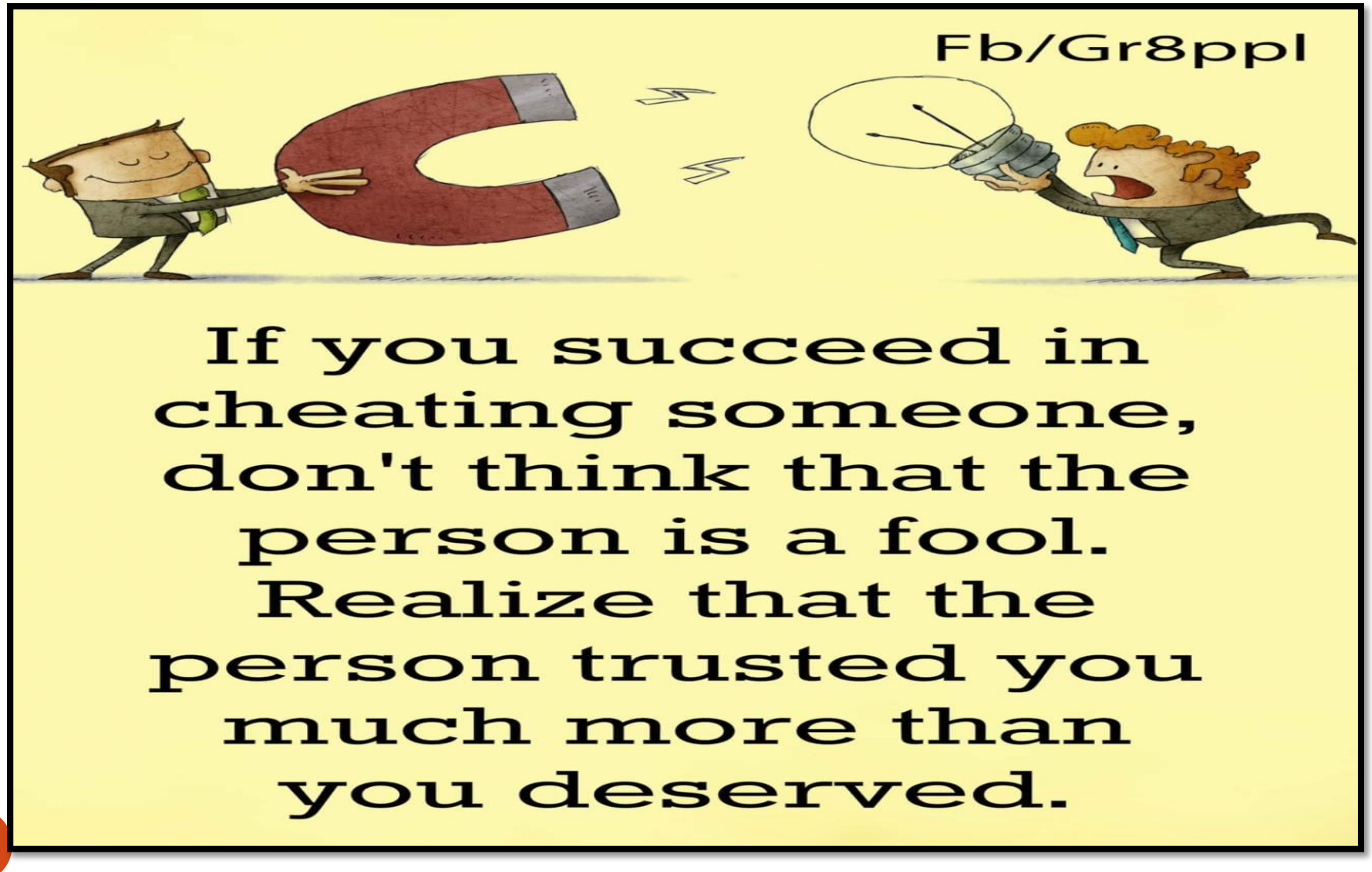
DropBox folder link

https://www.dropbox.com/sh/85vnrgkfqgrzhwn/AABdwKLJZqZs26a7u-y0AFwia?dl=0

Credits to Dr.Salma Hamdy for content preparation

Quote I of the Day-Midterm Related ☺

Quote II of the Day-Midterm Related ☺



Remark on 2D Array

2D array variable with one index

Hex representation of memory location of the i^{th} row, NOT the i^{th} element.

Examples

```
cout<<scores2D<<"\t\t"<<scores2D[0]<<"\t"<<scores2D[1]<<endl;
```

Displays location of the first element in first row, and location of the first element in the second row.

```
cout<<scores2D[0]<<"\t\t"<<scores2D[0]+1<<endl;
```

Display location of the first element, and location of the first element + 8 bytes (for double type).

Contents

Functions

Types of Functions

- **Built-In**
  - #include library file
  - Call the function giving its parameters
- **User Defined**
  - Function Prototype/Declaration
  - Function Definition/Body
  - Function Call

Example on Built-In Functions

- **Math Functions**
- **Strings functions**
- **Swap**
- **Rand**
- **File Streams (For Projects – not in the exam ☺)**

Built-in Functions (Mathematics)

Example 1:

Write a program that reads in 3 numbers and uses ***built-in functions*** for calculating the 1st number to the ***power*** of the 2nd number, and bring the ***square root*** of the 3rd number.

- **Sample execution:**

Enter the base: 3

Enter the exponent: 2

3 ^2 = 9

```
#include<iostream>
#include<math.h>    /*for built-in functions only add
the header file */
using namespace std;
void main()
{
float base, exponent, number, power, root;
cout<<"Enter the base: ";
cin>>base;
cout<<"Enter the exponent: ";
cin>>exponent;
power=pow(base,exponent);  /*calling*/
cout<<base<<"^"<<exponent<<"= "<<power<<endl;

cout<<"Enter number: ";
cin>>number;
root=sqrt(number);  /*calling*/
cout<<"The square root of "<<number<<" is "<<root<<endl;

}
```

Built-in Functions (Characters)

- **Example 2:**
- Write a program that counts the number of words and number of characters in a phrase typed by the user.

- **Hint**: Notice the difference between getch (reads char without showing it in the console screen) and getche (get character with echo)

```
#include<iostream>
using namespace std;
#include<conio.h>                    // for getche()
void main()
{
        int charcount=0;             //counts non-space characters
        int wordcount=0;             //counts spaces between words
        char ch;
        cout<<"Enter a phrase : ";
        ch = getche();               //reads 1st char to check on it in the loop
        while(ch!='\r')              //continue reading till you reach Enter
        {
           if(ch==' ')               //if a space is met, a new word is counted
             wordcount++;
           else
              charcount++;
           ch = getche();
        }
        if(charcount>0)
           wordcount++;   //last word is not counted in the loop because it's follwed by '\r' not
        space
        cout<<"\nnumber of characters = "<<charcount<<endl;
        cout<<"number of words = "<<wordcount<<endl;
}
```

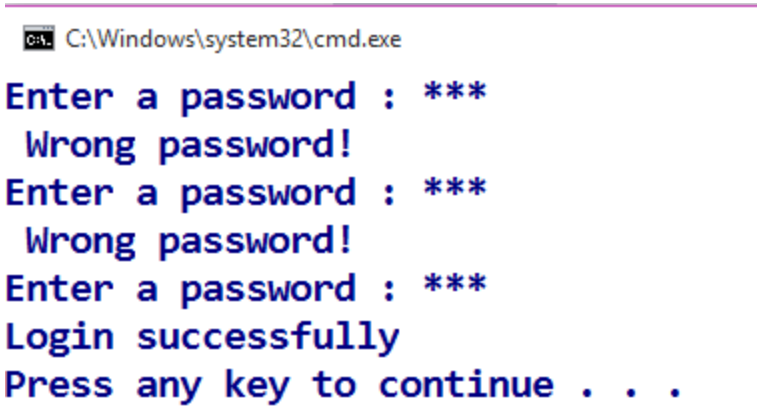
Example 3:

Write a program which asks the user to enter password, then it compares it to a stored password in the program and outputs valid or invalid. The user should be given only 3 trials then the program should terminate.

Note: As the user types the password on the screen, each character should appear as an asterisk.

(Hint: use **getch()** and do **NOT** use string library)

Code



Strings as an Array of characters

- We can deal with a string as an array of character.

- Any string must be terminated by **NULL** char **'\0'** as it is the only way any functions that work with a string will know where the string ends.

Declaration and Initialization

1D Arrays:

```
char  name1[4];                    //this size includes the null char
char  name2[] = {‘A’,’l’,’i’,’\0’};
char  name3[] = {“Ali”};  //the compiler will append ‘\0’
```

2D Arrays

```
char  name1[4][6];
char  name2[2][6]= {{‘A','l','i','\0'}, {'A','h','m','e','d','\0'}};
char  name3[][6] = {"Ali", "Ahmed"};
```

Input/Output with char arrays

Using for loops to read/print element by element just like any integer array Or ..

1D Array

```
char  name1[4];
gets(name1);
//cin>> name1;
cout<<name1;
```

To read a string consisting of one single word, as the >> operator considers a space to be a terminating character. While gets(stringname) reads till user presses enter

2D Array

```
char  name[2][6];
cin>>name[0]>>name[1];
cout<<name[0]<<‘  ‘<<name[1];
```

2 words each contains max 6 letters

Read 1st word then 2nd word

Print 1st word then 2nd word

Input1D array using built in functions

cin.get(CharArrayName1D, MaxArraySize);

Reads one or more characters from the input stream, its default delimiter is newline or if it reached MaxArraySize.

cin.get(CharArrayName1D, MaxSize, DelimiterChar);

To read multiple lines containing spaces and newlines as characters in the string. Its delimiter is user defined in the third parameter (any char) or if it reached MaxSize.

Example:

Input1D array using built in functions(cont.)

consider that our terminating char is ‘$’, then the function call will be like that:

cin.get(address, 20, ‘$’);

“Street 9
New Maadi$”

Note: This delimiter char will NOT be saved in the char array, it is just terminating character for the function.

Input1D array using built in functions(cont.)

Important note:

If you read a string with any **get** function after **cin>>,** the char array will read nothing and it will be an empty char array as the >> operator do not consume the newline, it just terminate, while **get** function consume the newline then terminate.

To avoid this conflict, you can use

cin.ignore()

Before using **get** after **cin>>**

that will consume the new line before taking the new input .

Built-in Functions (Strings)

gets: reads an array of characters terminated by a new line

puts: outputs an array of characters

strlen: returns the actual size of the string (without counting the null char)

strcpy: Copies string into another string

strcat: Concatenate strings

strcmp: Compares two strings

strchr: Locate first occurrence of character in string

strstr: Locate substring

strtok: Split string into tokens

strset: **S**ets a string to be all of a certain character

strupr: converts the string to uppercase

strlwr: converts the string to lowercase

strrev: reverses the characters of the string

int toupper(char c);

Convert single char c to upper case and return its new ASCII code

int tolower(char c);

Convert single char c to lower case and return its new ASCII code

Exercise

Example 4:

Write a program that uses the string operations to do the following:

Write a program that uses the string operations to do the following:

- **Read** 2 strings containing spaces and terminated by \n.
- **Find** the length of each string.
- **Compare** two strings.
- **Copy** a string into another one.
- **Set** all the characters in a given string with a given character.
- **Concatenate** a string into another.
- **Convert** a string to upper or lower case.
- **Reverse** a string.

Note: **gets** appends a null character \0 at the end of the string which is needed for all the string functions to work

```
#include<iostream>
using namespace std;
void main()
{
//definition
char str1[30],str2[60];
//reading using gets or while loop till \r using getche
//note: gets reads till enter
cout<<"Enter first string"<<endl; //note you must put endl
gets(str1);
cout<<"Enter second string"<<endl;
gets(str2);
//using strlen
cout<<endl;
cout<<"length of string1= "<<strlen(str1)<<endl;
cout<<"length of string2= "<<strlen(str2)<<endl;
cout<<"string1 is greater than string2"<<endl;
```

```
//using strcmp
cout<<endl;
int cmp=strcmp(str1,str2);
if(cmp==0)
   cout<<"the 2 strings are identical"<<endl;
else
   if(cmp<0)
      cout<<"string1 is less than string2"<<endl;
   else
      cout<<"string1 is greater than string2"<<endl;
//using strcpy
cout<<endl;
cout<<"After copying string1 to string2"<<endl;
strcpy(str2,str1);
cout<<"string2 has become ";//<<str2<<endl;
puts(str2);
```

```
//using strset (set all characters of a string to a given charactet)
cout<<endl;
char setchar;
cout<<"enter the character you want to convert string2 to: ";
cin>>setchar;
strset(str2,setchar);
cout<<"string2 has become "<<str2<<endl;
//using strcat
cout<<endl;
strcat(str1,str2);
cout<<"string1 after concatinating string2 : "<<str1<<endl;
//using strupr and strlwr (note:special characters are left as it is)
cout<<endl;
strupr(str1);
cout<<"string1 after converting it all to uppercase : "<<str1<<endl;
strlwr(str1);
cout<<"string1 after converting it all to lowercase : "<<str1<<endl;
//using strrev (reverses a given string)
cout<<endl;
strrev(str1);
cout<<"reversed string1 is : "<<str1<<endl;
```


TIME
for a
BREAK!

String library

As we said before, char array is also called string.

By including <string> library, we can use a pre-defined datatype called *string* which is exactly as character pointer or 1D character array.

Declaration and Initialization

```
string name;
string name1 = “Mohammed";
string name2(“Ali Ahmed");
string name3(name2);                    //name3 = name2
string text(3, ‘n’);                    //text = “nnn”;
```

Note:

Strings here do not need ‘\0’ in initialization the same as
char name3[] = {“Ali”}; //the compiler will append ‘\0’

Reading strings from user and Concatenating them

```
#include <iostream>
#include <string>
using namespace std;
void main()
{
    string name1;
    string name2;
    cin>>name1;                    // read string until the next separator
                                   // (space, newline, tab), not gets as it requires an array of chars
     getline (cin, name2);         // read a whole line into the string name2
    string name3 = name1 + ‘  ‘ + name2;
    cout<<name3<<endl;

}
```

String Operations (cont.)
Assignment, Append, and Swap

s1.assign(SourceString, StartIndexInSource, Count);

```
s1.assign(s2, 0, 3);
// assign the first three characters of s2 to s1
```

s1.append(SecondString, StartIndexInSource, Count);

```
s1.append(s2, 2, 3);
// append to s1 characters 2, 3 and 4 from s2
```

s1.swap(s2);

```
// exchanges s1 and s2
```

String Operations (cont.)
Insertion, Removal, and Replacement

s1.insert(StartPosition, SubString);

```
s1.insert(3, "abc");
// insert abc after position 3 in s1
```

s2.erase (StartIndex, Count)

```
s2.erase(4, 2);
// remove two characters starting from position 4
```

s3.replace(StartIndex, Count, SubString);

```
s3.replace(4, 2, "pqr");
// replace positions 4 and 5 with pqr
```

String Operations (cont.)
Copy and SubString

s1.copy(DestinationCharArray, Count, StartIndex);

s1.copy(s2,3,5);

// copy 3 characters from s1 into s2,

// starting after position 5 in s1

s1.substr(StartIndexOfSubString) ;

s1.substr(3);

// output substring starting from positions 3 to the end

s1.substr(StartIndexOfSubString, Count) ;

s1.substr(3, 2);

// output two characters starting from positions 3

String Operations (cont.)
Searching Operations

s1 = "mississippi";

```
s1.find("ss") ;
// returns 2
s1.find("ss", 3) ;
// returns 5

s1.rfind("ss") ;
// returns 5
s1.rfind("ss", 4) ;
// returns 2

i = s2.find_first_of("aeiou");
// find first vowel
//Searches the string for the first character that matches any of the characters specified in its arguments.
```

String Operations (cont.)
Comparison

```
string s1 = "hi ";
 string s2 = "world";
 if(s1 == s2)
     cout<<“s1 = s2"<<endl;
 else if(s1 > s2)
     cout<<"s1 > s2"<<endl;
 else
     cout<<"s1 < s2"<<endl;     //output will be s1<s2 as
                                //ASCII of ‘h’ is smaller
                                //than ASCII of ‘w’
```

ASCII Table

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

String Operations (cont.)
Character Access

```
cout << name[2] << endl;   // output position 2 of name
name[2]  =  'x';                          // change position 2 by ‘x’
cout << name.at(2) << endl;  // output position 2 of name
```

String Operations (cont.)
ASCII operations

```
string s1 = "abc";
s1[0] = s1[0] + 1;    //integer value will be added to
                              //the ASCII of the first char
s1[1] = s1[1] - 1;    //Subtract 1 from ASCII of 'b'
 s1[2] = s1[2] / 3;   //Divide ASCII of 'c' by 3
cout<<s1<<endl;

//output will be (according to the ASCII table):
ba!
```

Reading strings from user and Concatenating them

```
#include <iostream>
#include <string>
using namespace std;
void main()
{
    string name1;
    string name2;
    cin>>name1;                    // read string until the next separator
                                   // (space, newline, tab), not gets as it requires an array of chars
    getline (cin, name2);          // read a whole line into the string name2
    string name3 = name1 + ‘  ‘ + name2;
    cout<<name3<<endl;

}
```

TASK 4

- **Best Scorers..Thank you☺**
- **General (G1):**

1. **Abanoub Mouris**
2. **Ahmed Hossam**
3. **Rola Hanie**

TASK 4

- **Best Scorers…Thank you☺**
- **Software Engineering Department :**

1. **Magdy Ya’qob (Jacob☺)**
2. **Youssef Wael**
3. **Bashnona Gamal**

TASK 4

- **Best Scorers ..Thank You☺**
- **Bio-Informatics Department:**

1. **Demiana Emil**
2. **Nayera Mohamed Abd ElMeguid**
3. **Menna ALLAH Mohamed Foda**

TASK 4

- **Best Scorers..Thank You ☺**
- **General (G2):**

1. **Kareem Hany**
2. **Mahmoud Ashraf**
3. **Heba Gamal Saleh**

Wanted List

- **<u>The following students should meet me after the Lecture URGENTLY:</u>**

1. **Ahmed Rashad**
2. **Rana Moustafa**
3. **Abd-ALLAH Shehata**

- **Omneya Abd ElRahman**
- **Mohamed Taher Abd ElSamie**


ASSESSMENT

Midterm Exam

Midterm Content

1. **Theoretical : 1st lecture (and understanding how code works)**
2. **Practical: Lectures 1-6**
   1) **Control Structures (IF, Switch, Loops)**
   2) **Arrays (1D, 2D)**
   3) **Struct and Array of objects.**
   4) **Built-in functions** ***(In the exam you will be given its library and its prototype)***
3. **Questions type might include:**
   - **MCQ, Complete, True/false (Read carefully whether you are asked to correct the false statements)**
   - **Tracing the code and mention the output.**
   - **Converting PseudoCode to C++ code.**
   - **Fix code Errors (stating whether they are syntax, logical, runtime).**

Midterm Regulations

1. **DO NOT CHEAT – GOD is watching you ☺**
2. **Attend in your time slot.**
3. **WRITE YOUR FULL NAME IN ARABIC AS LISTED IN THE ATTENDANCE SHEET!**
4. **Come 10 mints before your exam start.**
5. **Have your own stuff (Pencils, pens, calculators…etc)**
6. **DO NOT TALK for any reason with anyone.**
7. **Have a question ask the TA, or myself.**
8. **Write your name on the exam paper.**
9. **YOUR MOBILE IS SHUT DOWN (Not silent or flight mode!) and not in front of you for any reason.**
10. **CLEAR and GOOD HANDWRITING Please ☺**

Midterm Schedule

Date	Time	Sections	Place
Wednesday 21/3/2018	8:00 am	Group 2 Sections: 17-22	Prof. Fahmy Tolba Lecture Hall
	9:00 am	Group 2 Sections: 12-16	
	10:00 am	Group 1 Sections: 7-11	
	11:00 am	Group 1 Sections: 1-6	
	Unspecified	Bio Informatics	Unspecified
	Unspecified	SW Engineering	Unspecified


Thank You