1. Lab 1:

Title: Familiarization with windows operating system and computer hardware

Objective:

- Enable students to be familiar with windows operating system
- Enable students to be familiar with the basic hardware components of computers

Preparation Tasks:

- -

Activities:

1.1. Familiarizing with windows operating system

1.2. Understanding directory structures, file/directory creation and deletion…

1.3. Running and closing application software

1.4. Editing of text documents using notepad and Microsoft word

2. Lab 2:

Title: Familiarization with Application software and Visual studio IDE

Objective:

- Familiarize students with visual studio IDE

Preparation Tasks:

- -
- -

Activities:

2.1. Running Microsoft visual studio IDE

2.2. Creating a new project and a new c++ file

2.3. Familiarizing the main components of visual studio software

2.4. Practicing how to compile and run simple c++ code using Microsoft visual studio

3. Lab 3:

Title: Familiarization with Visual studio IDE and executing simple codes

Objective:

- Enabling students to be familiar with different c++ program development IDEs [visual studio, code blocks, Devc++…)
- Enabling students to Execute simple codes
- Enabling students to debug programming bugs

Preparation Tasks:

- -

Activities:

3.1. Write a simple c++ program that displays a text to the console screen:

Welcome to AAiT computing LAB!"

3.2. Write a simple c++ program that takes the name of the user from key board and displays:

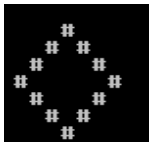
"Hello!

The name entered by user

Congratulations for being the part of this fantastic course!"

3.3. Write a C++ program that takes two numbers from the user and computes the sum, difference and product of the two numbers.

3.4. Write a simple c++ program that displays the following patterns at the console.



3.5. The following c++ program is excepted to display the following message:

My name is xyz

I am 24 years old

I have a great interest in c++ programming

But, unfortunately, the program has both syntax error and even after correcting the syntax error it doesn't display the expected text as per the requirement. So, identify the syntax bugs and debug them. In addition it is expected to correct the program to display the text as per the requirement.

```
#include<iostream>
Using namespace std;
Int Main ( )
{
      cout<<"my name is xyz"
      cout>>"I am 24 years old";
      cout<<"I have a great interest in c++ programming";
      Return 0;
}
```

3/22/2016

4. Lab 4:

Title: Using operators and library functions

Objective:

- Enable students to be familiar with the basic operators of C++ programming languages such as:
  - *Arithmetic operators*
  - *Relational and logical operators*
  - *Bitwise operators*
  - *Assignment and compound assignment operators*
  - *Increment and decrement operators and operator precedence.*
- Enable students to use basic c++ library functions during coding
  - *Mathematical functions*
  - *I/O manipulators*

Preparation Tasks:

- *Every* student *who come to the lab is expected to understand the exercises prior to attending the lab*
- *It is also expected to develop an algorithm for solving the laboratory exercises. The algorithm shall be developed either in pseudo code or flowcharts. Each student will submit its preparation at the end of the laboratory session and will be part of the laboratory exercise assessment. Those students who fail to do their preparation will be prohibited to attend the lab.*

Activities:

4.1. Write a simple c++ program that accepts a number from the user and displays the sine, cosine, tangent of that number.

4.2. Write a simple c++ program that calculates the area of any triangle using Heron's area formula.

Hint: Herons area formula

A=$\sqrt{s(s-a)(s-b)(s-c)}$

Where s=$\frac{a+b+c}{2}$ *and a, b, and c are the sides of any triangle.*

The program is expected to ask the user to enter the measure of the three sides of a triangle and display the result to the user in a user friendly manner.

The mechanism of taking input from user

Enter the first side of the triangle: ***shall be entered by user***
Enter the second side of the triangle: ***shall be entered by user***
Enter the third side of the triangle: ***shall be entered by user***

Expected Output

The area of the triangle is: ***the output calculated by the program***

4.3. Write a simple c++ program that accepts the values of two variables from the user and swaps the content of the variables.

4.4. Write a c++ program that calculate and display the total amount and the earned interest by a bank client for a certain amount of principal deposited in a bank.

The program is expected should interactive and take the required parameters from the user and should display the result to the console in a user friendly manner.

Hint: A=$P * (1 + \frac{R}{100})^T$

A=amount, p=principal, R=rate, T=time in year

The program is expected to take the following parameters from the user in the following manner

3/22/2016

⇨ **Mechanism of taking input from user**

Enter the principal: ***shall be entered by user***
Enter the Annual Rate of interest: ***shall be entered by user***
Enter the amount of time the principal is deposited in years: ***shall be entered by user***

⇨ **Expected Output**

The total amount of the deposit is: ***the output calculated by the program***
The net interest of the deposit is: ***the output calculated by the program***
Note: the result of the calculation shall have a maximum of two digit precision i.e. it should be like "the net interest of the deposit is: 1234.97 birr". To make the display more attractive it is expected to use the I/O manipulator of c++.

4.5. Modify the above program (4.4) for calculating of the principal deposited when the amount and other parameters are entered by the user.

5. Lab 5:

Title: Control Statements

Objective:

- *Students be able to describe the basic control statements of c++ language*
- *Effectively use the c++ control statements*
- *Effectively use the "while", " do…while", and "for loops" for writing c++ programs that solve real world problems*
- *Effectively use the conditionals("if" and nested "if….else" statements) and "switch" statements to control program flow control and solve real world problems*

Preparation Tasks:

- *Every student who come to the lab is expected to understand the exercises prior to attending the lab*
- *It is also expected to develop an algorithm for solving the laboratory exercises. The algorithm shall be developed either in pseudo code or flowcharts. Each student will submit its preparation at the end of the laboratory session and will be part of the laboratory exercise assessment. Those students who fail to do their preparation will be prohibited to attend the lab.*

Activities:

5.1. Write a c++ program that implements a console based calculator. The program shall prompt the user what kind of calculation he/she wants to perform and prints the result to the computer screen.

The program is expected to have the following functionalities:

1. Addition of two numbers:
2. Subtraction of two numbers:
3. Multiplication of two numbers:
4. Division of two numbers:
5. Display the trigonometric values of a number entered by user. The program is expected to display the sine, cosine, tangent, secant, cosecant and cotangent of a number entered by user.
6. The base ten logarithm and the natural logarithm of a number.

The features of the program.

- The program shall be user friendly and interactive.
- The program display shall be well formatted.
- The program shall be implemented using switch statements
- The program shall be reliable and error free i.e. the program shall display the appropriate error message if the user fails to enter the appropriate input. For example for division operation, if the denominator is zero, the program shall prompt the appropriate error message.

5.2. Write a c++ program that takes your birth year and reverse the order of the numbers to the console. If the user inters 1990 the program shall display 0991

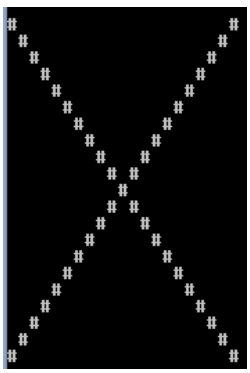
5.3. Trace the output of the following programs

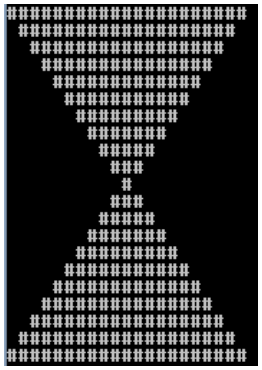
```
#include<iostream>
using namespace std;
int main()
  {
   int size;
   cout<<"enter the size"<<endl;
   cin>>size;
   Int c=0;

   for(inti=1; i<=size; i++)
   {
   while(true)
            {
                        cout<<"*";
                        c++;
                        if(c==(size+1)-i)
```

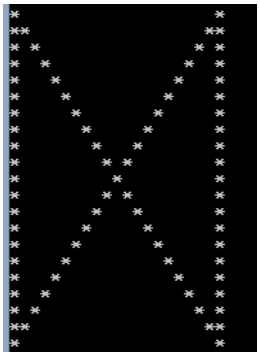
3/22/2016

```
                    break;
                }
                cout<<"\n";
            } //end of for loop
            return 0;
        } //end of main function
```

5.4. Using "loop" write a program that displays the following patterns. The size of the patterns shall be determined by the user.







5.5. Write a c++ program that sums the first n prime numbers. The number n should be entered by the user. The program execution should be:

What numbers of prime numbers you want to add: ***n will be entered by user***
*The sum of the first n prime numbers is:****n will be the program output***

5.6. Write a c++ that implements a number guessing game based on the following rules:

Rule 1: *The game has three guessing levels based on the first trial*
Level 1: *if the player guesses ± 5 of the number*
Level 2: *if the player guesses ± 15 of the number*
Level 3: *if the player guesses ± 25 of the number*

Rule 2: *The guessing chances will be as follows*
Level 1: *guess has 9 more chances to repeat*
Level 2: *guess has 6 more chances to repeat*
Level 3: *guess has 3 chances to repeat and all the chances of repletion's are allocated based on the first trial*

Rule 3: *If the player finds the number it shall prompt a message "congratulations you get the number the number is: the assigned number"*

Rule 4: *if the number is not found, the program shall give a hint of the interval in which the number is found. The interval shall be [assigned number-100, assigned number+100] and the hints will be useful for repeating the above steps for one more time only.*

Rule 5: *The program shall have a score report feature based on the following rules.*
If a person gets at its first trial its score will be 100
If the player guesses at level one in each trial 2poinntd will be deducted from 100
If the player guesses at level two in each trial 4 points will be deducted from 100
If the player guesses at level one in each trial 8 points will be deducted from 100
And finally based on the score the program is expected to deliver the percentage of the precession of the player.

5.7. Write a program that computes the product of the first 1000 numbers which are multiple of 7 using "while" loop only.

6. Lab 6:

Title: Arrays and Strings

Objective:

- *Students be able to explain basic data structures of c++*
  - *Understanding and using the basic data structures like arrays and strings to solve real world problem*
- *Students be able to explain single and multi-dimensional arrays*
- *Students be able to explain string manipulations*

Preparation Tasks:

- *Every student who come to the lab is expected to understand the exercises prior to attending the lab*
- *It is also expected to develop an algorithm for solving the laboratory exercises if the lab exercise requires. The algorithm shall be developed either in pseudo code or flowcharts. Each student will submit its preparation at the end of the laboratory session and will be part of the laboratory exercise assessment. Those students who fail to do their preparation will not to attend the lab session.*

Activities:

6.1. Write a c++ program that converts binary numbers in to the equivalent decimal numbers. The binary numbers shall be taken as single dimensional array from the user.

6.2. Write a c++ program that accepts a two 3X3 matrices and compute the following

a. Add the entire elements of the matrix and display to the console.

b. Store the sum of the entire rows of the first matrix in a single dimensional array

c. Transposes the two matrices

d. Computes the sum of the two matrices

e. Multiply the two matrices

f. Exchanges the diagonal of one of the matrices

6.3. Write a c++ program that that takes your name from the keyboard and displays the following: if your name has more than 10 characters, “your name is too long!” If it comprises between 5 and 9 characters, “your name is medium!” If it comprises less than five characters, “your name is to short!”

6.4. Write a c++ program that take to strings from keyboard and concatenate the two strings and store in the third string.

6.5. Write a c++ code that searches the term apply and country from the following text.

We offer a wide range of courses at all levels. If you are applying through one of our representatives in your country, they will guide you through making your application.

6.6. Write a c++ program that takes two strings from the keyboard and sorts them.

6.7. Write a c++ that take two integer values from the keyboard, compute their sum and convert back the result to string and concatenate the result with other string variable and print it to the console.

7. Lab 7:

Title: Structures and enumerated data types

Objective:

- *Understanding data structures of c++*
  - *Familiarize custom data types like enumeration, structures*
  - *Instantiating structures and accessing each element of the record in the main function*
- *Defining structures within structures*
- *Array of Structures*

Preparation Tasks:

- *Every student who come to the lab is expected to understand the exercises prior to attending the lab*
- *It is also expected to develop the high level structures diagram of the structures defined in 7.3*

Activities:

7.1. Define a structures called "Employee_Record" that records the information of typical employee in given company. The defined structures is expected to record the following employee data:

Employee's first name, middle name, last name, employee's ID, Gender, Employee's address (city, street, phone (mobile, fixed line for home and office, Fax, email, P.O.Box), department of work, salary, hire day and quit (date, month, year)

Note:

1. Each record element shall be assigned with the appropriate data type
2. The constructed structures shall be instantiated at the main function and the sample data type shall be feed to the record.
3. The records of the structure shall be displayed to the console with the appropriate formatting.

7.2. Define array of type Employee_Record with the size of 10 elements to store the records of the employees

7.3. Optimize the above structure defined in 7.1 by grouping the related data types within sub structures. Within the **Employee_Record** define the following structures and repeat the steps of 7.1 from 1-3.

Employee_name_rec which records the first name, middle name and last name of employee

Employee_adress_rec which records the city, street, mailing_tyepe_rec

The **mailing_type_rec** is a structures which records mobile, fixed line for home and office, Fax, email, P.O.Box of employee

Campany_specific_rec which records the salary, department of work, hire_date_rec, and quit_date_rec

Hire_date_rec and quit_date rec are structures that record the respective hire and quit day, month, and year of employee

Finally define structures called **Employee_Record** which records ***Employee_name_rec, Employee_adress_rec, Campany_specific_rec,*** **Hire_date_rec and quit_date_rec**

8. Lab 8:

Title: Functions

Objective:

- *Understanding the details of c++ functions*
- *Code modularization using functions*
- *Functions prototypes*
- *Understanding overloaded functions*
- *understanding recursive functions*

Preparation Tasks:

- *Every student who come to the lab is expected to understand the exercises prior to attending the lab*
- *It is also expected to develop an algorithm for solving the laboratory exercises if the laboratory exercises require. The algorithm shall be developed either in pseudo code or flowcharts. Each student will submit its preparation at the end of the laboratory session and will be part of the laboratory exercise assessment. Those students who fail to do their preparation will be prohibited from attending the lab.*

Activities:

8.1. Write a function that accepts an array of non-negative integers and returns the second largest integer in the array. Return -1 if there is no second largest integer.

8.2. Write a function that takes an array of integers as an argument and returns a value based on the sums of even and odd numbers in the array. Let X = the sum of the odd numbers in the array and let Y = the sum of the even numbers. The function should return X – Y.

8.3. Write a simple recursive functions that computes the factorial of a number and Fibonacci series of the number. The size of the number shall be determined by the user of the program.

8.4. Write a c++ function named prev_nxt that takes a number from the user and stores the predecessor and successor of that number in two previously defined variables. Those variables which are nominated for storing the predecessor and successor of the number shall be declared and initialized prior to the function call. The content of these variables shall be completely modified after the prev_nxt function call.

8.5. Write a c++ program that is supposed to perform addition of two integers, addition of two long integers, addition of two double precision floating numerals and concatenation of two strings (For this case you shall use the <string> library). All the above functionalities shall have their own c++ functions and all the functions shall be named **ovr_fun**.

8.6. Write a single c++ function called getMax which returns the bigger number among two integers, two double precision floating numerals and two single precision floating numerals. The function also expected to return a string or character which come at higher position when the two characters and strings are sorted alphabetically. The functions is required to return negative 1 (-1) if the two parameters are equal.

*(**Note**: For the case of numerals, the function returns the bigger number among the two if it exists but for character and strings the function is expected to return the one that come at higher level when they are sorted in alphabetical order)*

8.7. Write a ***c++ inline function*** that computes the sum of the first 1000 positive integer numerals recursively.

8.8. Assume that you are working in a large software company and participate in a certain software development project. The software, which is under development, has several modules and the modules are further divided into several sub modules. As a student and starter programmer, you are requested to develop the password validator sub module of the software based on the following rule. A password is valid if and only if the following criteria are fulfilled:

1. The password shall comprises combination of letters, numbers and special characters.
2. The length of the password shall be at least 15 characters.
3. The password shall have palindrome characteristics.

*(A palindrome is a string that reads the same forwards as backwards. The following are palindrome strings: **145787541, MADAM, RACECAR, ABABA, 1221**)*

The program shall be well organized and modularized through extensive use of c++ functions as much as the program complexity requires.

9. Lab 9:

Title: Pointers dynamic memory allocation memory management and File I/O

Objective:

- Understand the basic concepts of pointers
- Understand dynamic memory allocation
- Memory management concepts
- File I/0

Preparation Tasks:

- *Every student who come to the lab is expected to understand the exercises prior to attending the lab*
- *It is also expected to develop an algorithm for solving the laboratory exercises. The algorithm shall be developed either in pseudo code or flowcharts. Each student will submit its preparation at the end of the laboratory session and will be part of the laboratory exercise assessment. Those students who fail to do their preparation will be prohibited to attend the lab.*

Activities:

9.1. Write a c++ program that identifies that sorts an array. The size of the array shall be dynamically determined by the user at run time. The array elements either may be taken from the user or may be generated using c++ pseudo random number generator.

9.2. Write a c++ program that performs the following matrix operations.
Add two matrices, multiply two matrices and transpose a matrix.

- The program shall be user friendly and has a feature of correcting errors. For instance if a user tries to multiply two matrices of which the size of the row of one of the matrix is not the same as size of the column of another matrix shall throw exception and the exception shall be caught properly.
- The size of the matrices shall be determined by the user.
  1. All the operation shall be organized using functions. For all of the above operations, use dynamic memory allocation and a function pointer concept is mandatory.
  2. All dynamically allocated memory shall be freed after the use (use of memory management concept is mandatory)

9.3. Write c++ program that write the following text to the file called "myInfo.txt" at your preference directory of your computer in your computer.

```
||      My personal info:                ||
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
|       Name: your name                  |
|       ID. Number: your ID number       |
|       Hobby: your hobby                |
|       Additional info: any info        |
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

9.4. Update the existing text file you write in 9.3 to include your phone number, your country and your favorite book. You shall use I/O manipulators to format your text.

9.5. Write a c++ program which reads the text file that you write in 9.3 and 9.4 and displays it at the console.

9.6. Save the output of the programs that you did 9.1 and 9.2 in a text file in your computer at your preferred directory.

10. Lab 10:

Title: introduction to Classes and Inheritance

Objective:

- Introduce the basic Object-Oriented Programming concepts
- Understand the basic concepts of classes and objects in c++ programming language context

Preparation Tasks:

- *Every student who come to the lab is expected to understand the exercises prior to attending the lab*
- *It is also expected to develop an algorithm for solving the laboratory exercises. The algorithm shall be developed either in pseudo code or flowcharts. Each student will submit its preparation at the end of the laboratory session and will be part of the laboratory exercise assessment. Those students who fail to do their preparation will be prohibited to attend the lab.*

Activities:

10.1. Define a class called **RectangleCalculator.** The class shall have:

- Two private fields: **length and width**
- Functions that access the private fields of the class: setLength, setWidth, getLength, getWidth
- Constructors that initialize the fields of the class. The class shall have both parameterized and parameter less constructors.
- Destructors that free the memory from the constructed object when the created object becomes no serviceable in the program.
- Functions that calculate the area and perimeter of a rectangle

After defining the class it is expected to instantiate the class in the main functions and by feeding the required data test the required output.

10.2. Modify the above program by making all the constructors, destructors and functions implementation outside the class definition. (Hint: define the function prototypes in the class and use scope resolution operator and implement the functions outside the class definition.

10.3. (Optional) Define a class called parallelogramCalculator that inherits the class RectangleCalculator. The class shall have:

- One private field: **angle**
- Function that access the private field of the class: setAngle, getAngle
- Contractor that initialize the fields of the class (the constructors are expected to call the respective base class constructor)
- Functions that calculate the area of parallelogram (The program is expected to override (reemployment) the base class method (function)).

[Note: since the class is inherited from the RectangleCalculator, it is not expected to define functions and fields to get and set the length and width of the parallelogram rather it shall inherit them from its parent class.]

I HOPE WE HAVE SPENT GOOD TIME THROUGHOUT THE SEMISTER AND I WISH ALL GOOD THINGS TO BE WITH YOU IN ALL ASPECTS OF YOUR FUTUR LIFE!

GOOD LUCK!!!