THE MOMBASA POLYTECHNIC UNIVERSITY COLLEGE

FACULTY OF ENGINEERING &TEHNOLOGY

Department of Mechanical & Automotive Engineering

BACHELOR OF SCIENCE IN MECHANICAL ENGINERING (BSC YR2 SEM2)

Second Year Semester TWO Exam

Nov/Dec 2011

Computer programming for engineers (Matlab, Visual Basic, LabView, Object oriented C++)

Code: EMG 2210

Time 2 Hours

Instructions

Answer Question ONE & any other TWO questions

QUESTION ONE [COMPULSORY, 30 MARKS]

a) Describe any five features' of the LAB view application program [10 marks]

1. *LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where the flow of data determines execution.*

2. *LabVIEW builds a user interface by using a set of tools and objects.The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects.The block diagram contains this code. In some ways, the block diagram*
3. *resembles a flowchart.*

4. *You can purchase several add-on software toolsets for developing specialized applications. All the toolsets integrate seamlessly in LabVIEW.*

5. *LabVIEW is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and data acquisition control, vision, and motion control devices.*

6. *LabVIEW also has built-in features for connecting your application to the Internet using the LabVIEW web server and software standards such as TCP/IP networking and ActiveX.*

7. *LabVIEW can be used to create 32-bit compiled applications that give you the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions. You also can create stand-alone executables and shared libraries, like DLLs, because LabVIEW is a true*
8. *32-bit compiler.*

9. *LabVIEW contains comprehensive libraries for data collection, analysis,presentation, and storage. LabVIEW also includes traditional program development tools. You can set breakpoints, animate program execution,*

b) Write a C++ program that reads 100 numbers from the user and output their sum

[5 marks]

```
#include <iostream.h>
void main() {
int i, sum, x;
sum=0;
i=1;
while (i <= 100) {
        cin >> x;
        sum = sum + x;
        i = i+1;
}
cout << "sum is " << sum << endl;
```

c) Distinguish between a constructor and a destructor and show with code how each can be declared for a class the class above [4 marks]

Constructor: special function which is automatically called whenever a new object of this class is created
The destructor fulfills the opposite functionality. It is automatically called when an object is destroyed,

```
Circle::Circle (int a, int b) {}

Circle::~Circle () {}
```

d)Describe Four steps in object oriented design [8 marks]

I. *.Identify the candidate classes. List the nouns and noun phrases from the specification:*
II. *Identify the candidate operations. List the verbs from the specification.*
III. *Eliminate unnecessary and synonym classes and operations.*
IV. *Associate the operations with the appropriate classes.*

e) List three outputs of the object-oriented design phase*:* [3 marks]

i. *descriptions of each class,*
ii. *descriptions of each operation (i.e., method),*
iii. *diagrams giving relationships among the classes*

QUESTION TWO [20 marks]

a) Describe the applications of MATLAB program in engineering [5 marks]
Matlab (short for MATrix LABoratory) is a language for technical computing, developed by the The Mathworks, Inc. It provides a single platform for computation, visualization, programming and software development. All problems and solutions in Matlab are expressed in notation used in linear algebra and essentially involve operations using matrices and vectors. In *addition, you can use Matlab to build Graphical User Interfaces (GUIs) so that you can develop user-friendly custom software.*

b) A vector has four elements (a, b, c, d). Show how this can be created the follwing in matlab

i. Row vector
ii. Column vector
iii. Transpose [3 marks]

```
>> r = [a b c d]
>> c = [a; b; c; d]
>> c = r'
```

c) Represent the following two sets of matrices in matlab form [4 marks]

A=

1	2	3
4	5	6
7	8	9
10	11	12

B=

0	2	4	6	8	10
1	3	5	7	9	11

```
>> a = [1 2 3; 4 5 6; 7 8 9; 10 11 12]
```

*>> **b = [0:2:10; 1:2:11]***

d) Problem solving in matlab

i. Give a matlab expression for solving the following set of equations [5 marks]

a1 x + b1 y + c1 z = d1
a2 x + b2 y + c2 z = d2
a3 x + b3 y + c3 z = d3

we set this up as a matrix equation of the form
P U = Q
where
***P** = [a1 b1 c1; a2 b2 c2; a3 b3 c3]*
***U** = [x; y; z]*
***Q** = [d1; d2; d3]*

*The solution of this system of equations is **U = P^{-1} Q**; this is accomplished in Matlab using*
*>> **U = inv(P)*Q***

or by using the backslash \ operator
*>> **U = P\Q***

ii. Show the output the following matlab loop [3 marks]

```
>> for i = 1:10;
>> a(i) = i*i;
>> end
>> a
a =
```

1 4 9 16 25 36 49 64 81 100

QUESTION THREE [20 marks]

a)Dynamic memory allocation

In order to request dynamic memory we use the operator new. new is followed by a data type specifier and -if a sequence of more than one element is required- the number of these within brackets `[]`. It returns a pointer to the beginning of the new block of memory allocated. Its form is:

```
pointer = new type
pointer = new type [number_of_elements]
```

The first expression is used to allocate memory to contain one single element of type `type`. The second one is used to assign a block (an array) of elements of type `type`, where `number_of_elements` is an integer value representing the amount of these. For example:

```
1     int * bobby;
2     bobby = new int [5];
```

b) Write a C++ program that calculates the perimeter of a circle of radius 5

[5 marks]

```
#include <iostream>
using namespace std;
#define PI 3.14159
#define NEWLINE '\n'
int main ()
{
double r=5.0; // radius
double circle;
circle = 2 * PI * r;
cout << circle;
cout << NEWLINE;
return 0;
}
```

c)Write a program that outputs the following: 1, 2,3, FIRE! Using

i. A while loop
ii. A for loop [6 marks]

```
#include <iostream>
using namespace std;
int main ()
{
int n;
cout << "Enter the starting number > ";
cin >> n;
while (n>0) {
cout << n << ", ";
--n;
}
cout << "FIRE!\n";
return 0;
}
```

```
#include <iostream>
using namespace std;
int main ()
{
for (int n=3; n>0; n--) {
cout << n << ", ";
}
cout << "FIRE!\n";
return 0;
          }
```

d) List four benefits of objected oriented programming [4 marks]

1. *Software Reuse*
2. *Code Sharing*
3. *Improved Reliability*
4. *Easy maintenance*
5. *Rapid Prototyping*

e) State the meaning of the access specifiers, public, and private and protected [3 marks]

- *`private` members of a class are accessible only from within other members of the same class or from their friends.*
- *`protected` members are accessible from members of their same class and from their friends, but also from members of their derived classes.*
- *`public` members are accessible from anywhere where the object is visible.*

QUESTION FOUR [20 marks]

a) List four characteristics of a function [4 marks]

• Type is the data type specifier of the data returned by the function.
• Name is the identifier by which it will be possible to call the function.
• Parameters (as many as needed): Each parameter consists of a data type specifier followed by anidentifier, a regular local variable.
• Statements is the function's body. It is a block of statements surrounded by braces { }.

b) Differentiate between passing parameters by value and by reference [6 marks]

When a parameter is passed by value, a copy of the parameter is made. Therefore, changes made to the formal parameter by the called function have no effect on the corresponding actual parameter. For example:

```
void f(int n) {
   n++;
}
int main() {
   int x = 2;
        f(x);
        cout << x;
     }
```

When a parameter is passed by reference, conceptually, the actual parameter itself is passed (and just given a new name -- the name of the corresponding formal parameter). Therefore, any changes made to the formal parameter do affect the actual parameter. For example:

```
void f(int &n) {
   n++;
}
int main() {
   int x = 2;
   f(x);
   cout << x;
}
```

c) Write a c++ program that uses a function prototype to get the product of two numbers [5 marks]

```
#include <iostream>
using namespace std;
int mult ( int x, int y );
int main()
{
  int x;
  int y;

  cout<<"Please input two numbers to be multiplied: ";
  cin>> x >> y;
    cout<<"The product of your two numbers is "<< mult ( x, y ) <<"\n";
  }

int mult ( int x, int y )
```

```
{
  return x * y;
}
```

d) Demonstrate the concept of overloaded functions using a code snippet [5 marks]

Two different functions having the same name but different parameter types. eg

```
int operate (int a, int b)
{
return (a*b);
}
float operate (float a, float b)
{
return (a/b);
}
int main ()
{
int x=5,y=2;
float n=5.0,m=2.0;
cout << operate (x,y);
cout << operate (n,m);
}
```

QUESTION FIVE [20 marks]

a) Define the following terms

i. Class
ii. Object
iii. Method
iv. Abstraction [8 marks]

An object is an instance of a class. It can be uniquely identified by its name and it defines a state which is represented by the values of its attributes at a particular time.

A class is the implementation of an abstract data type (ADT). It defines attributes and methods which implement the data structure and operations of the ADT, respectively

A message is a request to an object to invoke one of its methods. A message therefore contains the name of the method and the arguments of the method.

Abstraction:hiding what is not needed. Showing only what is relevant

b) Write a C++ program that uses a class called Cylinder to calculate the volume of a sphere [6 marks]

```
#include <iostream>
Define pi 3.14
using namespace std;
class Cylinder{
int r,h;
public:
void set_values (int,int);
int volume () {return (4/3*pi*r*h);}
};
void Cylinder::set_values (int a, int b) {
r= a;
h= b;
}
int main () {
Cylinder  vol;
vol.set_values (3,4);
cout << "the volume is: " << vol.volume();
return 0;
}
```

c) Given that the class circle above is subclasses of a class called polygon, demonstrate with code snippet how inheritance can be implemented [6 marks]

```
class Polygon {
protected:
int width, height;
public:
void set_values (int a, int b)
{width=a; height=b;}
};
class Cylinder: public Polygon {
public:
int area ()
{ return (width * height); }
};
int main () {
Cylinder vol;volumet.set_values (4,5);
cout << volume.vol() << endl;
return 0;
}
```