Group Two

2

Group members..

- Bekoe Isaac
- Nana Boateng
- Emmanuel Sarfoh
- Dodoo Fredrick
- Odame George
- Prince Zormelo
- Joseph Verlan Kaya

Becwin-i
iziqbek@gmail.com

Group 2

Question1A.

```
#include <iostream>
using namespace std;

int main()
{
   string Exam_type,student_score,Quiz,mid_term,final_score,name;
   float score_1,score_2,score_3,average_score;
   cout<<"Enter the score of the Quiz test_ ";
   cin>>score_1;
   cout<<"Enter the score of the Mid-Semester test_ ";
   cin>>score_2;
   cout<<"Enter the score of the Final test_ ";
   cin>>score_3;

    if (score_1>=90 && score_1<=100)
      {
      cout<<"Exam"<<"  "<<"Score"<<"  "<<"Grade"<<endl;
      cout<<"Quiz"<< "\t" << score_1 <<"\t"<<"A"<<endl;
      }
      else if (score_1>=70 && score_1<=90)
      {
      cout<<"Exam"<<"  "<<"Score"<<"  "<<"Grade"<<endl;
      cout<<"Quiz"<< "\t" << score_1 <<"\t"<<"B"<<endl;
      }
      else if (score_1>=50 && score_1<=70)
```

Group 2

```
{
cout<<"Exam"<<  "<<"Score"<<  "<<"Grade"<<endl;
cout<<"Quiz"<< "\t" << score_1 <<"\t"<<"C"<<endl;
}
else if (score_1>=0 && score_1<=50)
{
cout<<"Exam"<<  "<<"Score"<<  "<<"Grade"<<endl;
cout<<"Quiz"<< "\t" << score_1 <<"\t"<<"F"<<endl;
}
else
{
   cout<<"ambiguous score entered"<<endl;
}
if (score_2<=100 && score_2>=90)
{
cout<<"Mid-sem"<< "\t" << score_2 <<"\t"<<"A"<<endl;
}
else if (score_2<=90 && score_2>=70)
{
cout<<"Mid-sem"<< "\t" << score_2 <<"\t"<<"B"<<endl;
}
else if (score_2<=50 && score_2>=50)
{
cout<<"Mid-sem"<< "\t" << score_2 <<"\t"<<"C"<<endl;
}
else if (score_2<=50 && score_2>=0)
```

Group 2

```
{
cout<<"Mid-sem"<< "\t" << score_1 <<"\t"<<"F"<<endl;
}
else
{
   cout<<"ambiguous score entered"<<endl;
}

 if (score_3<=100 && score_3>=90)
{
cout<<"Final"<< "\t" << score_3 <<"\t"<<"A"<<endl;
}
else if (score_3<=90 && score_3>=70)
{
cout<<"Final"<< "\t" << score_3 <<"\t"<<"B"<<endl;
}
else if (score_3<=50 && score_3>=50)
{
cout<<"Final"<< "\t" << score_3 <<"\t"<<"C"<<endl;
}
else if (score_3<=50 && score_3>=0)
{
cout<<"Final"<< "\t" << score_3 <<"\t"<<"F"<<endl;
}
else
{
```

Group 2

```
   cout<<"ambiguous score entered"<<endl;
}
cout<<endl;
cout<<"_____"<<endl;
average_score=(score_1+score_2+score_3)/3;
 if (average_score<=100 && average_score>=90)
{
cout<<"Average score = "<<average_score<<" is a grade A"<<endl;
}
else if (average_score<=90 && average_score>=70)
{
cout<<"Average score = "<<average_score<<" is a grade B"<<endl;
}
else if (average_score<=50 && average_score>=50)
{
cout<<"Average score = "<<average_score<<" is a grade C"<<endl;
}
else if (average_score<=50 && average_score>=0)
{
  cout<<"Average score = "<<average_score<<" is a grade F"<<endl;
}
else
{
   cout<<"ambiguous average score entered"<<endl;
}
```

```
    return 0;
}
```

Program output

```
Enter the score of the Quiz test_ 99
Enter the score of the Mid-Semester test_ 87
Enter the score of the Final test_ 76
Exam  Score  Grade
Quiz    99      A
Mid-sem 87      B
Final   76      B

_____________________________________
Average score = 87.3333 is a grade B

Process returned 0 (0x0)   execution time : 6.039 s
Press any key to continue.
```

Group 2

Presentation

Question 1b. The program below has a syntax mistakes.

a. Identify the mistakes

b. Explain why the mistake

c. Correct the mistakes The program

Original program

```
# include < iostream >
const int a = 11.213
const b = 15.6
int main ()
{
 int k, L;
int m, n;
L = 7;
 m = 3.00;
L = L + n;
a = L + a;
cout << a << endl;
c = b * 36.75;
 cout << “Wages = ” << c << endl;
return 0;
}
```

Corrected program

```
#include<iostream>
using namespace std;
const float a=15.6;
const float b=11.213;
int main ()
{
 float k, L,c;
 float m, n;
 L = 7;
 m = 3.00;
 L = 0;
n = 0;
 cout << a << endl;
 c = b * 36.75;
 cout << "Wages = " << c << endl;
 return 0;
 }
```

✓ **Identifying the mistakes.**

1. “using namespace std;” wasn’t declared.
2. Wrong spacing of “#include <iostream>”
3. Wrong variable definition.
4. const b lacks a variable definition.
5. Both const a and b didn’t end with a semicolon.
6. Wrong variable definition for k, L ,m & n.
7. Based on the formula in line 10 & 11
8. n=0 & L=0

9. c wasn't declared

- ✓ **Explaining the identified mistakes.**

"using namespace std;" wasn't declared.-

The built in C++ library routines are kept in the standard namespace. That includes stuff like cout, cin, string, vector, map, etc. Because these tools are used so commonly, it's popular to add "using namespace std" at the top of your source code so that you won't have to type the std:: **prefix** constantly.

Wrong spacing of "#include <iostream>"

"#include <iostream>" is the preprocessor directive and its case sensitive and cant' be executed since it has wrong spacing.

Wrong variable definition.-

Since the value of const a & b are decimals the variable definition should be **float** not **int.**

const b lacks a variable definition-

Every variable has needs a variable definition before it can be use. example of variable definitions are int, float, double etc.

Both const a and b didn't end with a semicolon-

Every c++ statement has to end with semicolon (;) to end the statement. Therefore, a & b wasn't ended since it has no semicolon at the end of the statement.

Wrong variable definition for k, L, m & n.-

Since the program has decimals, every variable must be declared with float to avoid output of inaccurate answers.

Based on the formula in line 10 & 11, n=0 & b=0. –

L= L + n

n=L – L

Therefore, n=0.

a=L + a

L=a – a

Therefore, L=0.

From the equation above n & L can be declared as 0.

c wasn't declared-

Since c wasn't declared and can't be processed.

✓ **Correcting the identified mistakes.**

- "using namespace std;" has to be typed under "#include<iostream>" since it defines which namespace to use in the program.
- "#include <iostream>" must be retype with one spacing between the #include and the <iostream>.
- Wrong variable definition can be corrected by changing the int to float.
- const b must have a float variable in front of b = 15.6 .
- End const a and const b with a semicolon to end the statement.
- Variable definition of k, L, m and n must be replaced with float.
- The formula L=L + n and a=L + a must be deleted and replaced with n=0 and L=0.
- c must be declared using the variable definition of float.

Question 2A.

```
#include <iostream>


using namespace std;


int main()
{
   int x,x1,a,b,c,z;
   cout<<"Input the values of a,b and c respectively"<<endl;
   cin>>a>>b>>c;
   if(a==0 && b==0)
   {
      cout<<"No solution"<<endl;
   }
   else if(a==0)
   {
```

```
    x=-c/b;
    cout<<c<<" / "<<b<<"= "<<x<<endl;
  }
  else if(a<0 && b<0 && c<0)
  {
    cout<<"No root"<<endl;
  }
  else
  {
  x1=b+(((b^2)-4*a*c)^1/2)/2*a;
  x=b-(((b^2)-4*a*c)^1/2)/2*a;
  cout<<"x1 = "<<x1<<endl;
  cout<<"x  = "<<x<<endl;
  }
  return 0;
}
```

Program output

```
Input the values of a,b and c respectively
0
2
4
4 / 2= -2
process returned 0 (0x0)   execution time : 3.039 s
Press any key to continue.
```

Question2B.

```
#include <iostream>


using namespace std;


int main()
{
   float p,q,n;
   p = 10.0;
   q = 15.0;
   n = 6.0;
   cout<<"p = "<<p<<", q = "<<q<<", n = "<<n<<endl;
   cout<<"p + n = "<<p+n<<endl;
   cout<<"sum of "<<p<<" and "<<n<<" is "<<p+n<<endl;
   return 0;
}
```

Program output

```
p = 10, q = 15, n = 6
p + n = 16
sum of 10 and 6 is 16


Process returned 0 (0x0)   execution
time : 0.015 s
Press any key to continue.
```

Group 2

Question 2C. Three control structures in high level language programming in the following:

- 'if'
- If else
- If else if

Describe how each of this operates.

How 'if' works in high level programming.

The meaning of TRUE and FALSE in computer terminology. A true statement is one that evaluates to a nonzero number. A false statement evaluates to zero. When you perform comparison with the relational operators, the operator will return 1 if the comparison is true, or 0 if the comparison is false. For example, the check 0 == 2 evaluates to 0. The check 2 == 2 evaluates to a 1. If this confuses you, try to use a cout statement to output the result of those various comparisons (for example cout<< (2 == 1);)

When programming, the aim of the program will often require the checking of one value stored by a variable against another value to determine whether one is larger, smaller, or equal to the other.

There are a number of operators that allow these checks.

Here are the relational operators, as they are known, along with examples:

```
>    greater than            5 > 4 is TRUE
<    less than               4 < 5 is TRUE
>=   greater than or equal    4 >= 4 is TRUE
<=   less than or equal       3 <= 4 is TRUE
==   equal to                5 == 5 is TRUE
!=   not equal to            5 != 4 is TRUE
```

How 'if else' works in high level programming.

An **if** statement can be followed by an optional **else if...else** statement, which is very usefull to test various conditions using single if...else if statement.

When using if , else if , else statements there are few points to keep in mind.

- An if can have zero or one else's and it must come after any else if's.
- An if can have zero to many else if's and they must come before the else.
- Once an else if succeeds, none of he remaining else if's or else's will be tested.

Syntax

The syntax of an if...else if...else statement in C++ is −

```
if(boolean_expression 1) {
   // Executes when the boolean expression 1 is true
} else if( boolean_expression 2) {
   // Executes when the boolean expression 2 is true
} else if( boolean_expression 3) {
   // Executes when the boolean expression 3 is true
} else {
   // executes when the none of the above condition is true.
}
```

How 'if else if' works in high level programming.

if...else if...else Statement

An **if** statement can be followed by an optional **else if...else** statement, which is very usefull to test various conditions using single if...else if statement.

When using if , else if , else statements there are few points to keep in mind.

- An if can have zero or one else's and it must come after any else if's.
- An if can have zero to many else if's and they must come before the else.
- Once an else if succeeds, none of he remaining else if's or else's will be tested.

Syntax

The syntax of an if...else if...else statement in C++ is –

```
if(boolean_expression 1) {
   // Executes when the boolean expression 1 is true
} else if( boolean_expression 2) {
   // Executes when the boolean expression 2 is true
} else if( boolean_expression 3) {
   // Executes when the boolean expression 3 is true
} else {
   // executes when the none of the above condition is true.}
```

Question 3A.

```
#include <iostream>


using namespace std;


int main()
{
   float score_1,score_2,score_3,score_4,score_5,Quiz_Total,Total;
   cout<<"Results:==========Quizzes==========="<<endl;
   cout<<"Enter the score of the first quiz: ";
   cin>>score_1;
   cout<<"Enter the score of the second quiz: ";
   cin>>score_2;
   cout<<"Enter the score of the third quiz: ";
   cin>>score_3;
   cout<<"==========Mid-Term==========="<<endl;
   cout<<"Enter the score of the mid-term: ";
```

Group 2

```
    cin>>score_4;
    cout<<"==========Final==========="<<endl;
    cout<<"Enter the score of the final: ";
    cin>>score_5;
    Quiz_Total=score_1+score_2+score_3;
    cout<<"Quiz Total: "<<"\t"<<Quiz_Total<<endl;
    cout<<"Mid-Term: "<<"\t"<<score_4<<endl;
    cout<<"Final:     "<<"\t"<<score_5<<endl;
    cout<<"....................."<<endl;
    Total=Quiz_Total+score_4+score_5;
    cout<<"Total:     "<<"\t"<<Total<<endl;



    return 0;
}
```

Program output

```
Results:==========Quizzes==========

Enter the score of the first quiz: 45

Enter the score of the second quiz: 65

Enter the score of the third quiz: 77

==========Mid-Term==========

Enter the score of the mid-term: 76

==========Final==========

Enter the score of the final: 54

Quiz Total:     187

Mid-Term:       76

Final:          54

.....

Total:          317


Process returned 0 (0x0)   execution time : 11.967 s

Press any key to continue.
```

Question3B.i

Program requirement.

1.The score of first, second & third quiz.

2.The score of mid term.

3.The score of final test.

ii <u>An algorithm</u>

- Input the score of the first, second and third quiz.
- Input the score of the mid term test.
- Input the score of the final te
- Find the total score using the formula …
  Total score=(first quiz + second quiz + third quiz)+mid term score + final test score

Question 4A. Name a c++ reserve word/predefined identifier that allows for the following to be accomplished in a program. With an example, indicate how this reserve words are used.

i. create variables whose values don't change

ii. to output stream

iii. input streams

iv. end a statement and move the cursor to the next line

Group 2

i const is the function used to create a variable whose value doesn't change.

Example of the application of **const**,

```
#include<iostream>
using namespace std;
const float a=15.6;
const float b=11.213;
int main ()
{
 float c;
 float k, L;
 float m, n;
 L = 7;
 m = 3.00;
 L = 0;
 n = 0;
 cout << a << endl;
 c = b * 36.75;
 cout << "Wages = " << c << endl;
 return 0;
```

ii cout<<" "; is the command used to output streams.

Example of the application of **cout<<" ";**

```
#include <iostream>

using namespace std;

int main()

{

   cout << "Hello Mr.Freeman" <<
endl;

   return 0;

}
```

iii cin>> ; is the command use to input streams.

Example of the application of **cin>> ;**

```
#include <iostream>

using namespace std;

int main()

{

   string name;

   cout<<"Enter you name"<<endl;

   cin>>name;

   cout<<"Your name is
"<<name<<endl;

   return 0;

}
```

iv . endl; command is used end a statement and move the cursor to the next line.

Group 2

Example of the application of the **endl;** command,

```
#include <iostream>

using namespace std;

int main()

{

   string name;

   cout<<"Enter you name"<<endl;

   cin>>name;

   cout<<"Your name is
"<<name<<endl;

   return 0;

}
```

Question 4B. The area and perimeter of a circle can be determined using the following formula:

Area = pi r2 , where pi = 3.142 and r is radius.

Perimeter = 2 pi r, where pi = 3.142 and r is radius.

Assuming that a program was to be written to determine the Area and perimeter above, perform a program requirement analysis to determine the input, the process and output.

- Designing an algorithm

1. Input radius.
2. Get the value of Area using the following formula,
   - Area = pi *r*2, where pi = 3.142 and r is radius.
3. Get the value of pi.
   - where pi = 3.14
4. Get the value of perimeter using the formula,
   - Perimeter = 2* pi* r, where pi = 3.142 and r is radius.

Group 2

Question 4C. The While Loop is a repetition statement and the if statement is a condition statement. How different is repetition statement from a conditional statement?

Answer-

An **if statement** checks if an expression is **true** or **false**, and then runs the code inside the statement only if it is true. The code inside the **loop** only runs **once**.

A **While loop** is a loop that **continues to execute** the code in the **while statement** for however long the expression is **true**.

Question 5C. Suppose p, q, n, and t are double variables. What value is assigned to each of these variables after the last statement executes

```
p = 15.00;
q = 10.00;
n = p – 8;
t = p + 5 * q - n;
n = t – p;
t++;
```

Using c++ program analysis…….

```
#include <iostream>
using namespace std;
int main()
{
double p,q,n,t;
p=15.00;
q=10.00;
n=p-8;
t= p + 5 * q - n;
n= t - p;
t++;
cout<<"t = "<<t<<endl;
cout<<"n = "<<n<<endl;
cout<<"p = "<<p<<endl;
cout<<"q = "<<q<<endl;
   return 0;
}
```

Program Output dislplay…

```
t = 59
n = 43
p = 15
q = 10

Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
```

Group 2

Using c++ program programming, the values assigned to the variables are

t = 59

n = 43

p = 15

q = 10.

Question 5A.

```
#include <iostream>

using namespace std;

int main()
{
   float s,x,p,y,total;
   cout<<"Enter the values of s,x,p and y respectively_ "<<endl;
   cin>>s>>x>>p>>y;
   total=s*2+p*(s-x)*(p+y);
   cout<<"Total = "<<total<<endl;


   return 0;
}
```

Program output

```
Enter the values of s,x,p and y
respectively_
2
3
5
6
Total = -51


Process returned 0 (0x0)   execution
time : 2.811 s
Press any key to continue.
```

Question 5C. Differentiate between syntax and Semantics.

Answer-

Syntax is about the **structure** or the grammar of the language. It answers the question: how do I construct a valid sentence? All languages, even English and other human (aka "natural") languages have grammars, that is, rules that define whether or not the sentence is properly constructed.
Here are some C language syntax rules:

- separate statements with a semi-colon
- enclose the conditional expression of an IF statement inside parentheses
- group multiple statements into a single statement by enclosing in curly braces
- data types and variables must be declared before the first executable statement (this feature has been dropped in C99. C99 and latter allow mixed type declarations.)

Semantics is about the **meaning** of the sentence. It answers the questions: is this sentence valid? If so, what does the sentence mean
are syntactically valid C statements. But what do they mean? Is it even valid to attempt to transform these statements into an executable sequence of instructions? These questions are at the heart of semantics.

Consider the ++ operator in the first statement. First of all, is it even valid to attempt this?

- If x is a float data type, this statement has no meaning (according to the C language rules) and thus it is an error ***even though the statement is syntactically correct.***
- If x is a pointer to **some data type**, the meaning of the statement is to "add sizeof(**some data type**) to the value at address x and store the result into the location at address x".
- If x is a scalar, the meaning of the statement is "add one to the value at address x and store the result into the location at address x".

Finally, note that some semantics cannot be determined at compile-time and must therefore must be evaluated at run-time. In the ++ operator example, if x is already at the maximum value for its data type, what happens when you try to add 1 to it? Another example: what happens if your program attempts to dereference a pointer whose value is NULL?

In summary, syntax is the concept that concerns itself only whether or not the sentence is valid for the grammar of the language . Semantics is about whether or not the sentence has a valid meaning.

Question 6C. What role does the index of an array play?

The index of an array represents the number chosen in the list of numbers in the array elements.