STRAND	CE 13.3 APPLICATION PACKAGES
SUB STRAND	CE 13.3.2 PROGRAMMING
CONTENT LEARNING OUTCOME	CE 13.3.1.2 ANALYZE AND CONSTRUCT PROGRAMS USING PROGRAMMING SKILLS LEARNT FROM C++

Repetition statements

- The real power of the program is realized when the same type of statements can be executed over and over again.
- The repetition statements both define the boundary of the program containing the repeated section of the program and controls whether the code should be executed on not.
- In general there are two forms of repletion statements:
  1. While
  2. For

While Loops

- In C++ while loop is constructed using a **while** statement
- The syntax of the statement is as follows:

    while (expression)
    statement

- The expression contained within the parenthesis is the condition tested to determine if the statement following the parenthesis is to be executed.
- In the while statement the expression is executed repeatedly until the condition holds true.

Example 1

This program enables a user to enter English and Maths marks. The program will calculate and display the average mark. This program will run for 3 students.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main ()
{
int E,M,count,A;
count=0;
while (count<3)
{
cout<<"enter english mark"<<endl;
cin>>E;
        cout<<"enter maths mark"<<endl;
        cin>>M;
        A=(E+M)/2;
        cout<<"Average is"<<A<<endl;
        count=count+1;
        }
        system("pause");
        return (0);
        }
```

The output of the above program is shown below.

Example 2
This program displays all even numbers from 2 to 20.

```cpp
#include <iostream>
#include <stdlib.h>
using namespace std;

int main ()
{
 int count;
 count=2;
 while (count<=20)
 {
 cout<<count<<endl;
  count=count+2;
  }
  system("pause");
  return (0);
  }
```

The output produced by the above program is shown below.

Example 3
The program given below asks the user to answer a question. The relevant statements will be repeated until the user types in the right answer.

```cpp
#include <iostream>
#include <stdlib.h>
using namespace std;

int main ()
{
 int answer;
 while (answer !=15)
 {
```

```cpp
cout<<"What is the answer of the following   5 + 5 X 2"<<endl;
cin>>answer;
 }
 system("pause");
return (0);
}
```

Given below is the output produced by the above program.

In the example given so far we have used the **while** loop. Now we will look at the for loop.

For Loops

- In C++ for loop is constructed using for statement. The statement performs the similar function as while loop but uses different statement.
- For Loop is preferable in situations which uses a fixed count.
- The syntax of for loop is given is follows:
- **for(counter initialization; expression; counter alteration)**
   Statement;

EXAMPLE 4
The following program does exactly the same thing as Example 1 but uses FOR loop.

```cpp
#include <iostream>
#include <stdlib.h>
using namespace std;
int main()
{
float  eng,maths,average;
for (int i = 1; i< 5; i++)
{
cout<<"enter english mark"<<endl;
cin>>eng;
cout<<"enter maths mark"<<endl;
cin>>maths;
average=(eng+maths)/2;
```

```
cout<<"average= "<<average<< endl;
}
system("PAUSE");
return 0;
}
```
Shown below is the output of the above program.

```
enter english mark
90
enter maths mark
80
average= 85
enter english mark
70
enter maths mark
80
average= 75
enter english mark
60
enter maths mark
60
average= 60
Press any key to continue . . .
```

EXAMPLE 5

The following program does exactly the same thing as Example 2 but uses FOR loop.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main()
{
int num;
for (num = 2; num<= 20; num+=2)
{
cout<<num<<endl;
}
system("PAUSE");
return 0;
}
```
Output produced by the above program is as follows.

```
2
4
6
8
10
12
14
16
18
20
Press any key to continue . . .
```

Example 6

The program given below displays all multiples of 5 from 50 down to 5.

```
#include <iostream>
#include <stdlib.h>
using namespace std;
int main()
{
int num;
for(num = 50; num >=5; num-=5)
{
cout<<num<<endl;
}
system("PAUSE");
return 0;
}
```

Given below is the output of the above program.

```
50
45
40
35
30
25
20
15
10
5
Press any key to continue . . . _
```

ACTIVITIES

1. **Write a C++ program that will display your name 5 times.** **(4 marks)**

2. **Write a C++ program that will display all odd numbers from 1 to 31.** **(6 marks)**