**COMPUTER PROGRAMMING METHODOLOGY - BIT2201**

**CAT TAKE AWAY**

**MEMBERS**

**MECHUMO ISAAC - BSCAS/2020/92517**

**DAVIS GATI - BSCAS/2020/68006**

**NORAH BILHA - BSCAS/2020/68993**

**AUDREY ASHER -BSCAS/2020/67623**

**PHILIP TULATIA - BSCAS/2020/91761**

**CECILIA NZILANI-BSS/2018/86314**

**CHARLY STEPHEN -BSCAS/2020/92535**

**ROSE GICHOHI - BSCAS/2020/67841**

**GERALDINE NZIOKA - BSCAS/2020/90249**

**CHAPTER THREE**

**PROGRAM SRUCTURE**

A C program basically consists of the following parts:-

Functions

Variables

Statements and expressions

Comments

**CHAPTER FOUR**

**Basic syntax**

Tokens in c is either a keyword, identifier, a constant a string literal or a symbol. eg.

Semicolon : e.g printf(goodmorning);

return 0;

Comments : Are helping text in C program include. e.g.

/\* my first program in C \*/

Identifiers : is a name used to identify a variable function. e.g.

mynames,temp.etc

**Keywords**

whitespace in C;it describes blanks,tabs,newline,characters and comments.

**CHAPTER FIVE**

**Data types**

It refers to an extensive aystem used for declaring variables or function of different types.they include:-

**1.Basic types**

(i) Integer types

(ii) Floating point type.

(iii) Void type specifies no value is available.

**CHAPTER SIX.**

**VARIABLES**

Is a name given to a storage area that our programs can manipulate.

Examples;

char - An integer type.

int - Integer for machine..

float - single presicion floating point value

Double - a double presicfion floating point value.

Void - Represent the absence of type.

**Variable definition**

It tells the computer and how much storage to create for the variable**.**

**Examples are;**

extern int d = 3,f = 5 // declaration of d and f

int d = 4,f = 6 // definition and initializing d and f

**CHAPTER SEVEN**

**Constants and Literals**

Constants are fixed values that do not change during execution.They are also called literals.

Integer literals;can be a decimal,octal or hexadecimal constant e.g.

212 /\* legal \*/

54 /\* decimal \*/

Floating point literals has a integer part,a decimal point,a fractional part an exponent part

e.g.2.6457 /\* legal \*/

Character constant

Are enclosed in single quotes e.g.'x'

**Examples** that depict a few escape sequence characters are;

String literals

Are enclosed in double quotes e.g. "hello madam"

Defining constants

They include:- a).using#define preprocessor

b).using costant keyword

a)The #define preprocessor

b) The costant keyword

**CHAPTER EIGHT**

**Storage classes**

A storage class defines the defines the scope (visibility) and define variables and/ functions within a c program

**CHAPTER NINE**

Operators

Is a symbol that tells the compiler to perform specific mathematical or logic functions.

**1). Arithmetic operator**

**2). Relational operators;**

**3). Logical operators**

**4). Assignment operators.**

**CHAPTER TEN**

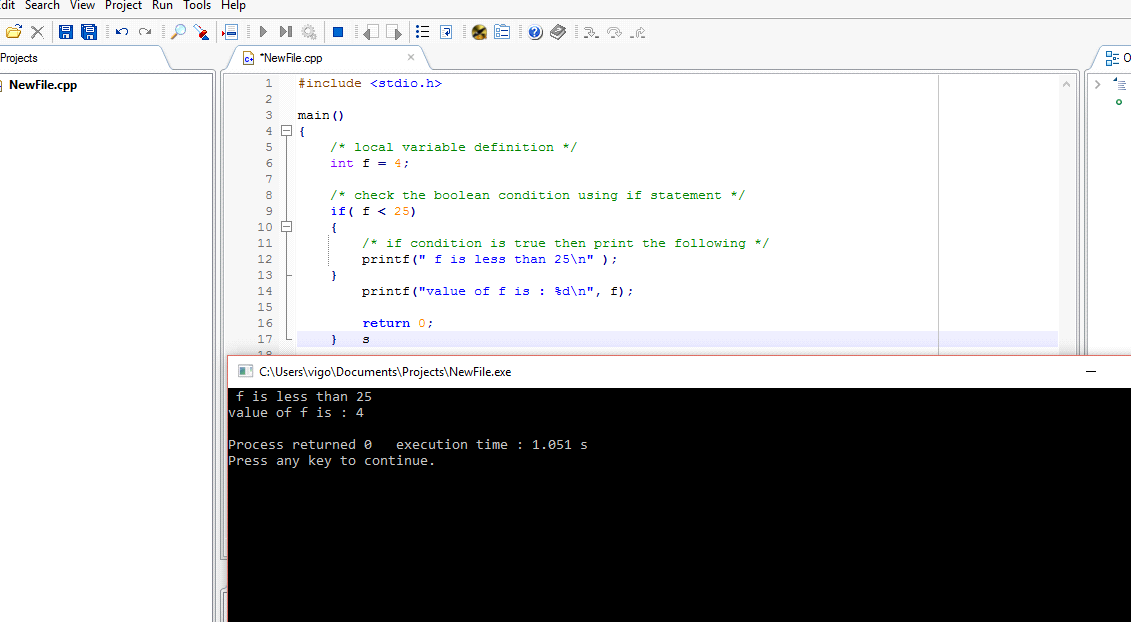
**Desicion making**

This requires that the programmer specifies one or more conditions to be evaluated or tested by the program.

They include the following;

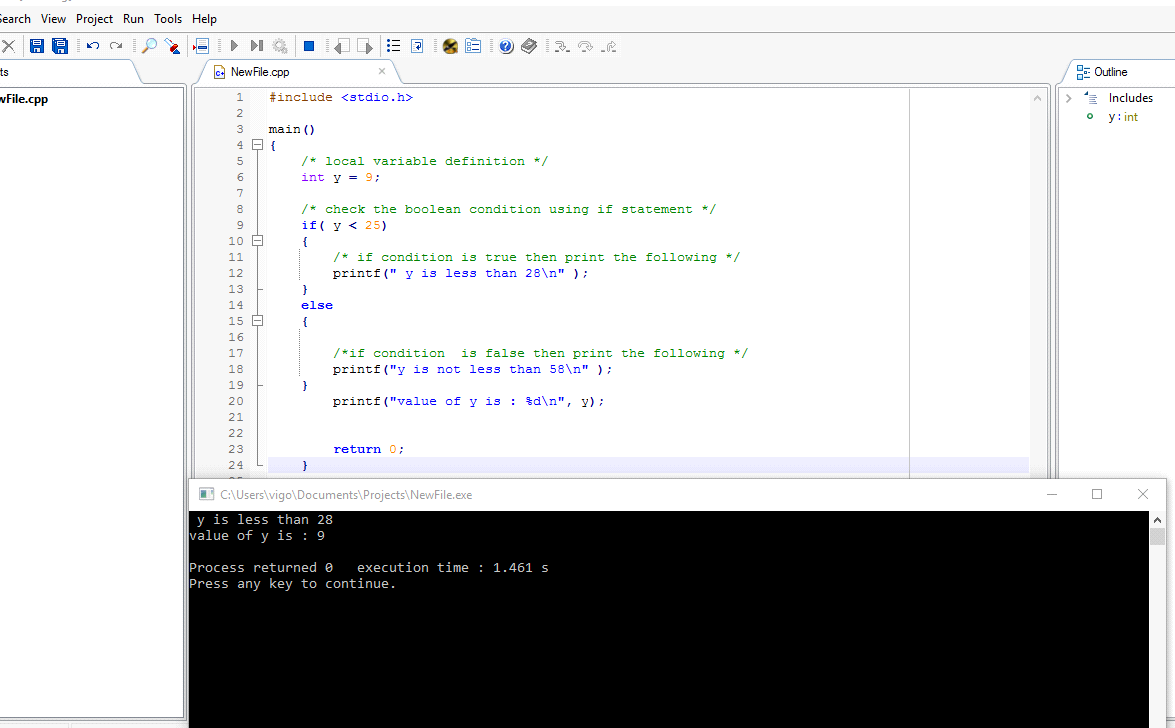
**i) if statement;**

**Example**;



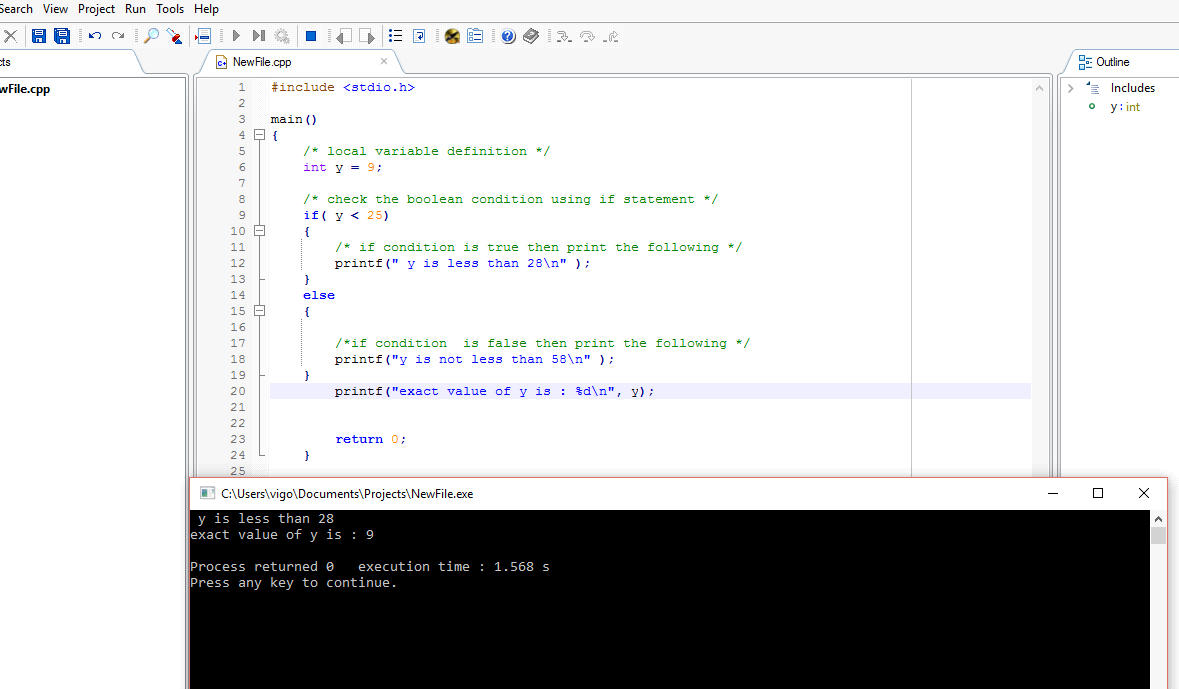
**ii). If.....else statement.**

**Example;**

****

**iii). IF else if......else statement.**

**Example;**

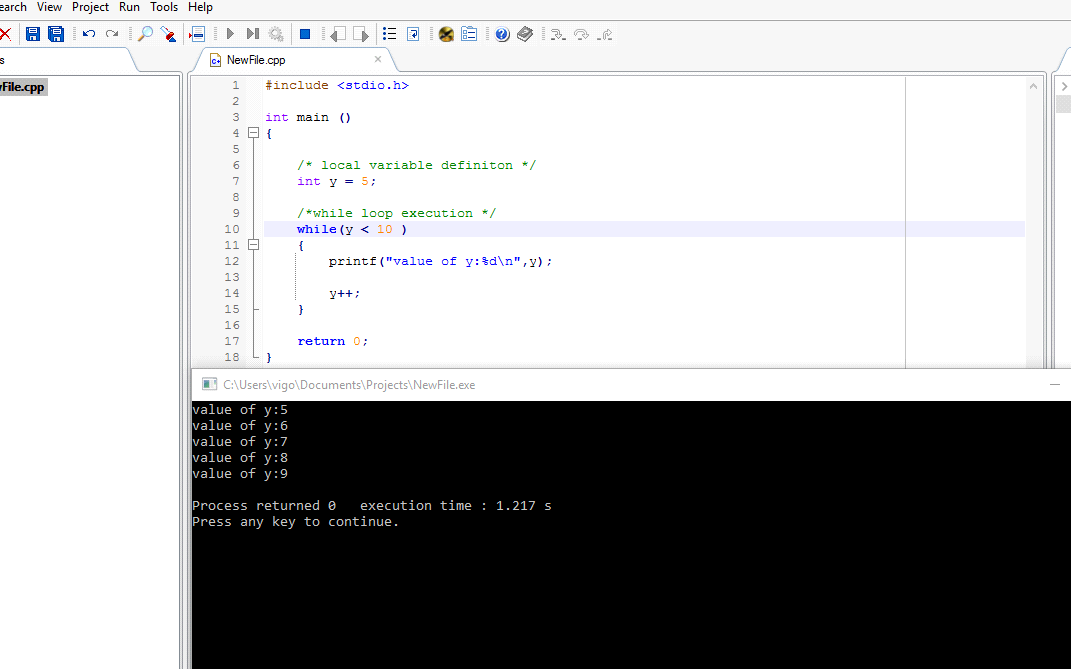
****

**CHAPTER ELEVEN.**

**loops;**

**1) While loop.** It executes a target statements as long as given a condition is true.

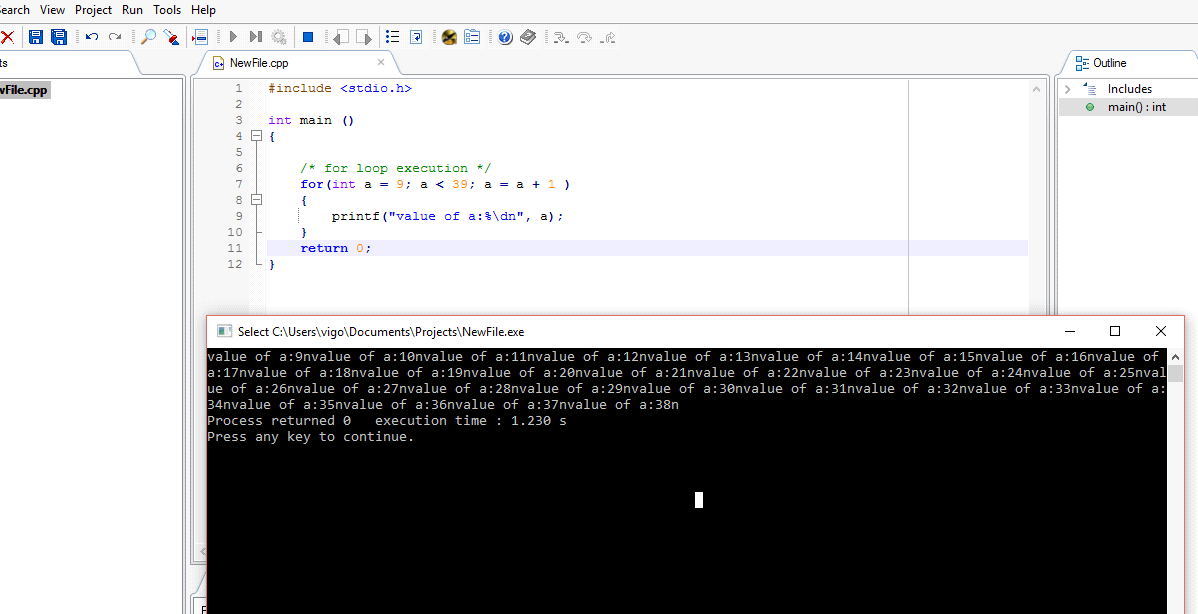
**Example**;



**2). for loop;**

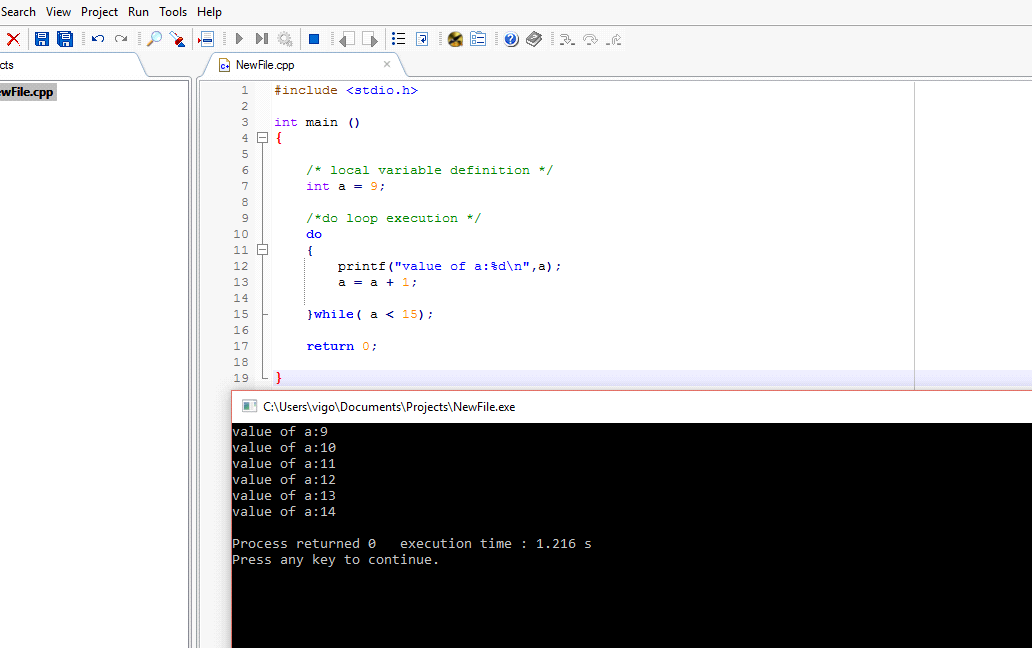
is a repetition control structures that allow you to efficiently write a loop that needs to execute a specific number of times.

**Example;**

****

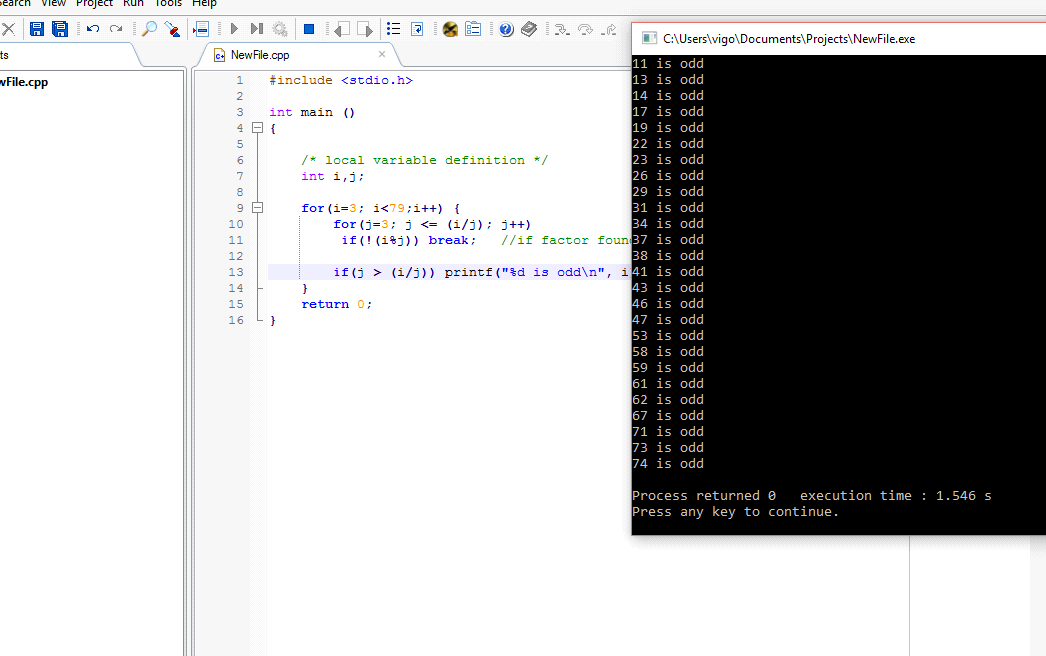
**3).do...while looop;**

**Example;**

****

**4). nested loops.**

**Example;**

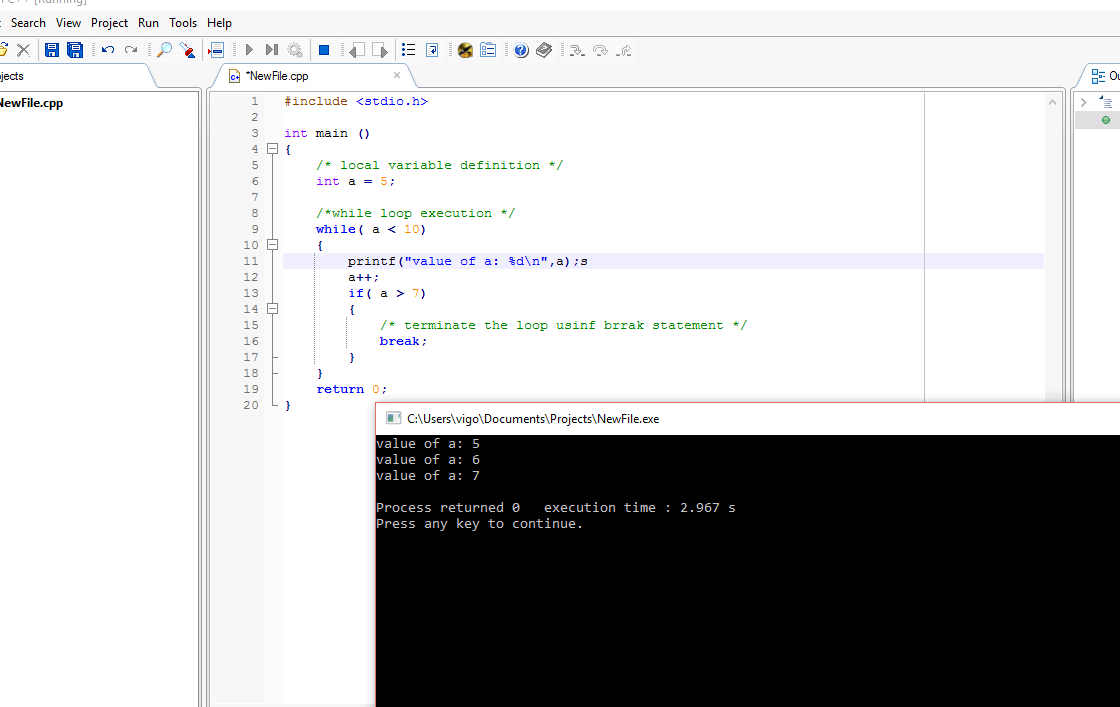
****

**5). loop control statements;**

c supports the following under this**;**

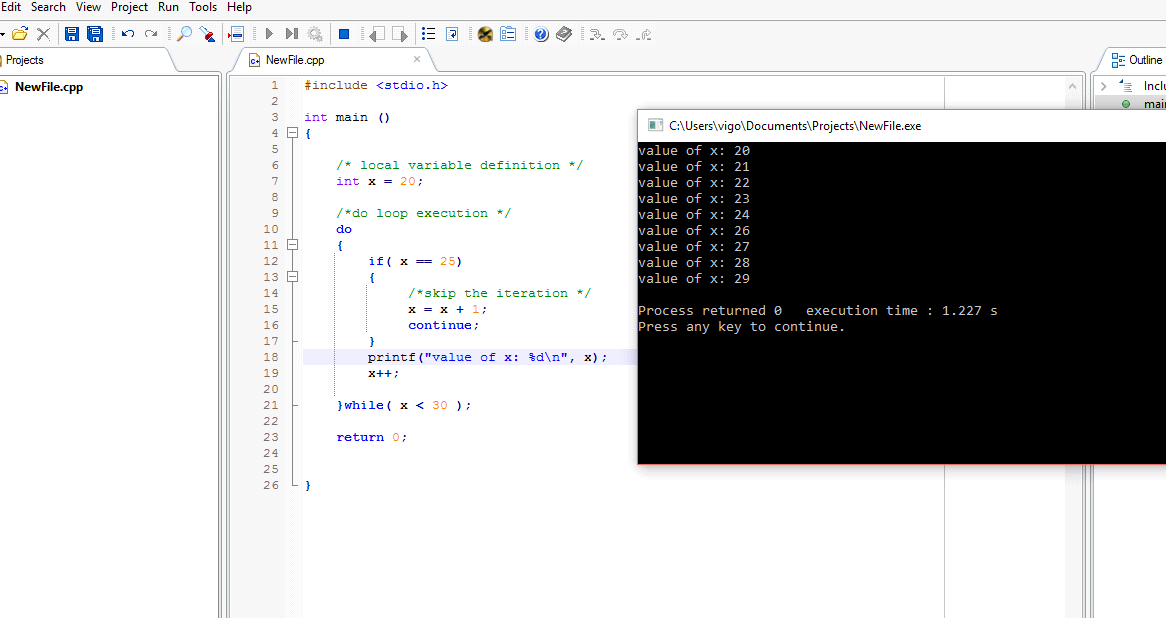
**a). break statement;**

**Examples;**

****

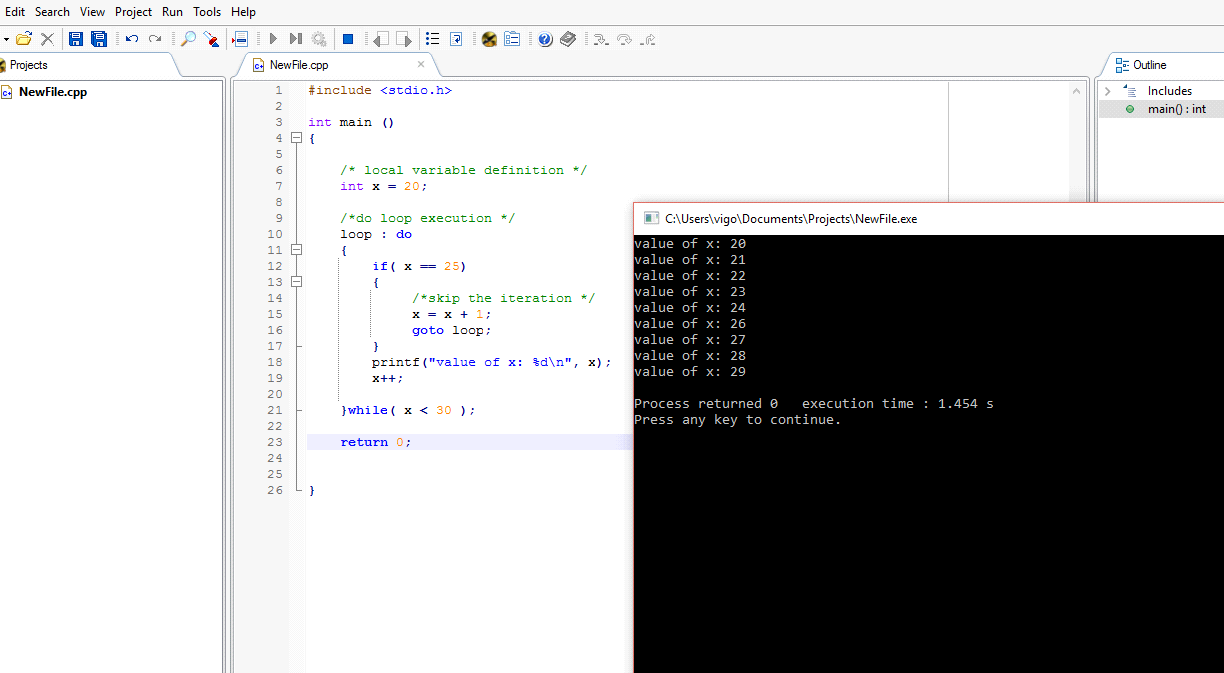
**b). continue statement;**

**Example;**

****

**c). goto statement**

**Example;**

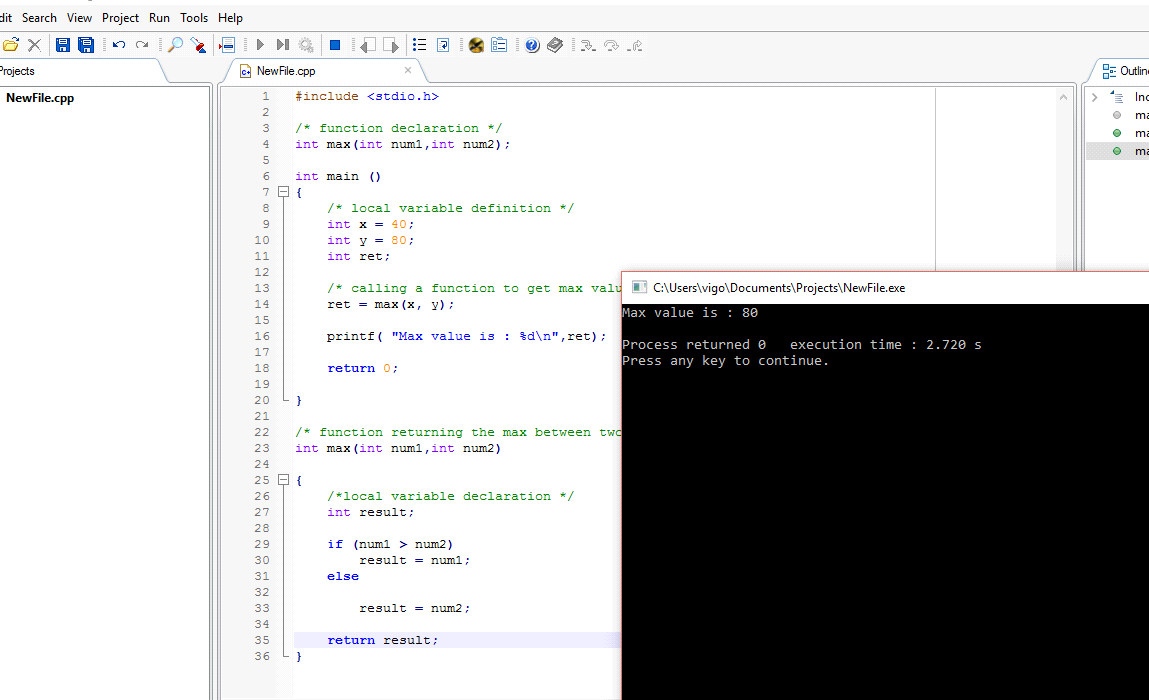
****

**CHAPTER TWELVE;**

**Functions**

**Calling functions**

**Example;**

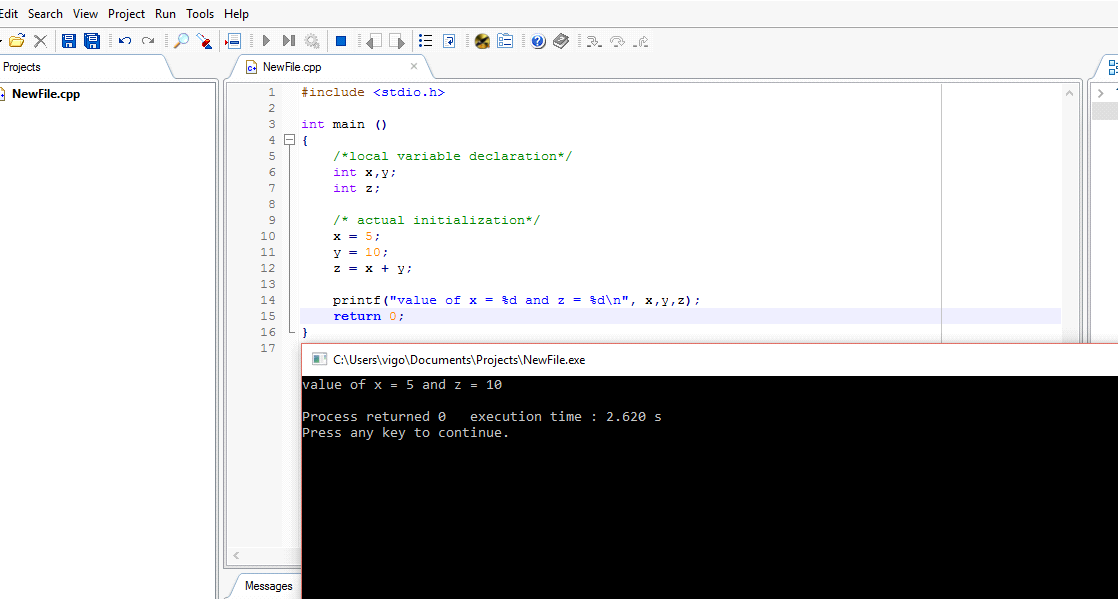
****

**CHAPTER THIRTEEN**

**Scope rules**

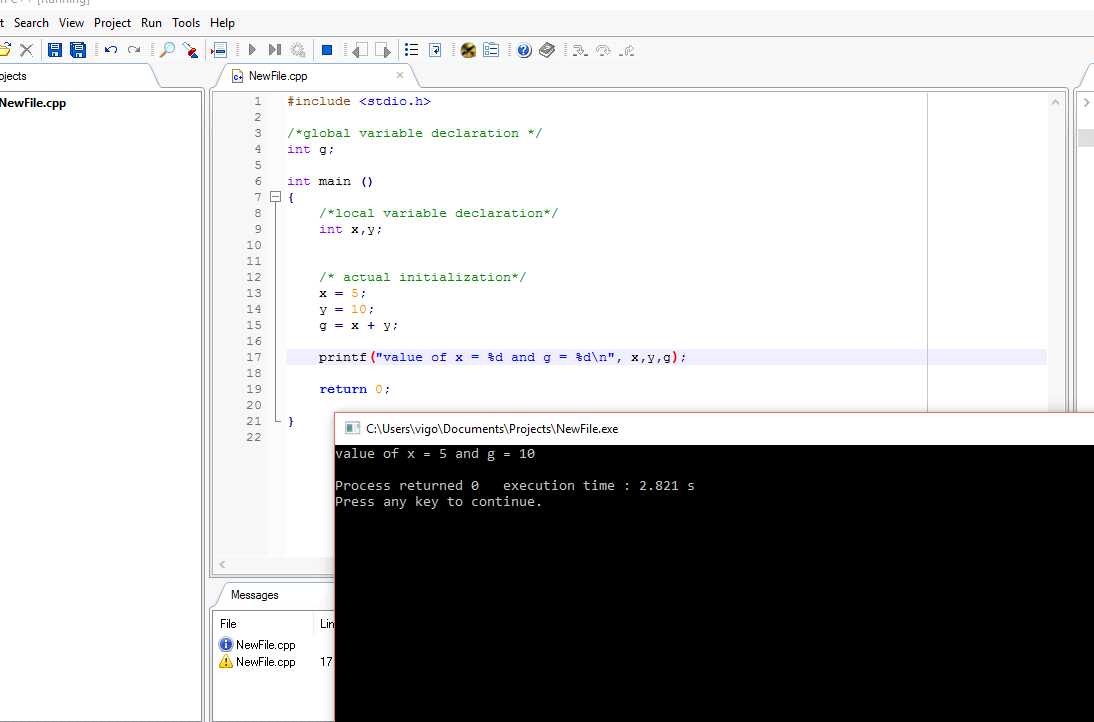
**Local variables**

**Example;**

****

**global variables**

**Examples;**

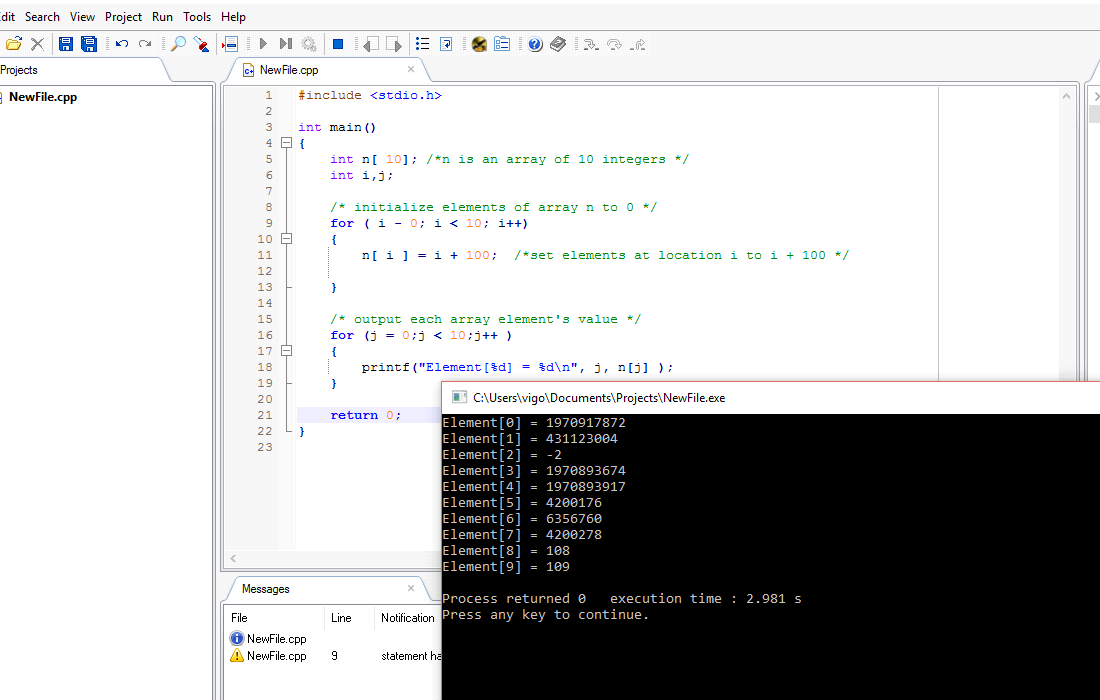
****

**CHAPTER FOURTEEN**

**Arrays**

Is a kind of data structure that can store a fixed size sequential collection of elements of the same type.

**Example of Accessing Array elements.**

****