MCAT113: Principles of Programming with C

Multiple choice questions

UNIT I

1. After a programmer plans the logic of a program, she will next ____.
a. understand the problem
b. test the program
c. translate the program
d. code the program

2 The process of walking through a program's logic on paper before you actually write the program is called ____.
a. desk-checking
b. flowcharting
c. pseudocoding
d. testing

3. Structured programs can be easily broken down into routines or ____ that can be assigned to any number of programmers.
a. segments
b. modules
c. units
d. sequences

4. The ____ can be a useful tool when a program must be modified months or years after the original writing.
a. flowchart
b. hierarchy chart
c. pseudocode
d. variable declaration

5. Structured Programming is used to describe
a. how programs are to be written
b. processes in a DFD in a reasonably precise manner
c. how DFDs are used in data processing
d. data flows in DFD

6. The objective of using structured Programming is to
a. describe computational procedures reasonably precisely which can be understood by any user
b. expand a DFD so that a user can understand it
c. develop algorithms corresponding to processes in a DFD
d. ease writing programs for DFDs

7. What is a compiler?
a. A compiler does a conversion line by line as the program is run
b. A compiler converts the whole of a higher level program code into machine code in one step
c. A compiler is a general purpose language providing very efficient execution

8. What programming languages are classified as
a. Basic, COBOL, FORTRAN,low-level languages? and .C.
b. Machine code and Assembly Languages
c. Prolog 2, Expert Systems, Knowledge Based Systems.

9 Fortran does not permit recursion because
a. It uses static allocation for storing variables
b. It uses dynamic allocation for storing variables
c. Stacks are not available in all machines
d. It is not possible to implement recursion on all machines

10. Which of the following language is case sensitive.

a. Forton
b. Basic
c. C
d. None

11. Programming Language offers features to write functions to

a. Facilitate to implementation of top-down logic
b. Enhance logical clarity
c. Avoid programming across program
d. None of above

12.You are asked to use to solve a problem given to you. How fast the computer solves

your problem, depends on the

a. Algorithm used
b. Language used for implementation
c. Programmer
d. Computer

13. Val is a well known
a. Real-time language
b. Object-oriented language
c. Command language
d. Data-flow language

14. The principle that a function can always be replaced by its value (irrespective of the context) without changing the meaning is called

a. Referential transparency
b. Orthogonality
c. Context-free
d. Unbinding

15. Which of the following comparisons between static and dynamic type checking is incorrect?

a. Dynamic type checking slows down execution
b. Dynamic type checking offers more flexibility to programmers
c. Dynamic type checking is more reliable
d. Unlike static type checking, dynamic type checking is done during compilation

16. How can we represent the logic of the program?

a. Program
b. Pseudocode
c. Algorithm
d. Flowchart

17. the period of time between an allocation and its subsequent disposal is called

a. Scope
b. Dynamic binding
c. Lifetime
d. Longevity

18. for which of the following applications will you prefer a co-routine to a subroutine?

a. Simulation of multi-processing
b. Complex searching process
c. Handling inter-leaved lists
d. None of the above

19. Binding cannot be done

a. When separately compiled modules are being linked together
b. During loading
c. While writing a program
d. None of the above

20. Fortran does not permit recursion because

d. It uses static allocation for storing variables
e. It uses dynamic allocation for storing variables
f. Stacks are not available in all machines
g. It is not possible to implement recursion on all machines

UNIT II

1. What will be output if you will compile and execute the following c code?

```
#define x 5+2
void main(){
int i;
i=x*x*x;
printf("%d",i);
}
```

a. 343

b. 27

c. 133

d. Compiler error

e. None of above

2. What will be output if you will compile and execute the following c code?

```
void main(){
int i=4,x;
x=++i + ++i + ++i;
printf("%d",x);
}
```

a. 21

b. 18

c. 12

d. Compiler error

e. None of above

3. What will be output if you will compile and execute the following c code?

```
void main(){
int a=10;
printf("%d %d %d",a,a++,++a);
```

}

a. 12 11 11

b. 12 10 10

c. 11 11 12

d. 10 10 12

e. Compiler error

4. C is which kind of language?

a. Machine
b. Procedural
c. Assembly
d. Object-oriented
e. Strictly-typed

5. In a C expression, how is a logical AND represented?

a. @@
b. ||
c. AND.
d. &&

6. Which one of the following C operators is right associative?

a. =
b. ,
c. []
d. ^
e. ->

7. How do you include a system header file called sysheader.h in a C source file?

a. #include <sysheader.h>
b. #incl "sysheader.h"
c. #includefile <sysheader>
d. #include sysheader.h
e. #incl <sysheader.h>

8. Which of the following are the legal C statements:

1) N='BBDNITM';
2) A=b=3=4;
3) d=b^2- 4*a*c;
4) si=p * r * t /100;

a. 1
b. 4
c. 3
d. 1 & 4

9. How do printf()'s format specifiers %e and %f differ in their treatment of floating-point numbers?

a. %e always displays an argument of type double in engineering notation; %f always displays an argument of type double in decimal notation.
b. %e expects a corresponding argument of type double; %f expects a corresponding argument of type float.
c. %e displays a double in engineering notation if the number is very small or very large. Otherwise, it behaves like %f and displays the number in decimal
d. %e displays an argument of type double with trailing zeros; %f never displays trailing zeros.

10. A C variable cannot start with

1. An alphabet
2. A number
3. A special symbol other than underscore
4. 2 & 3, both

a. 3
b. 4
c. 2
d. None

11. In the C language 'a’ represents

a. a digit
b. an integer
c. a character
d. a word

12. What is a difference between a declaration and a definition of a variable?

a. Both can occur multiple times, but a declaration must occur first.
b. There is no difference between them.
c. A definition occurs once, but a declaration may occur many times.
d. A declaration occurs once, but a definition may occur many times.

13. Which one of the following printf() format specifiers indicates to print a double value in decimal notation, left aligned in a 30-character field, to four (4) digits of precision?

a. %-30.4e
b. %4.30e
c. %-30.4f
d. %-4.30f

14. char txt [20] = "Hello world!\0";

How many bytes are allocated by the definition above?

a. 11 bytes
b. 20 bytes
c. 13 bytes
d. 12 bytes

15. short int x;

What is the maximum number that can be printed using printf("%d", x);

a. 127
b. 128
c. 32,767

d. 65,536

16. The expression 5<8<2 is equivalent to

a. 0
b. 1
c. 5
d. 2

17. What will be output if you will compile and execute the following c code?

```
void main(){
int x=5,y=10,z=15;
printf("%d %d %d");
 }
```

a. Garbage Garbage Garbage

b. 5 10 15

c. 15 10 5

d. Compiler error

e. Run time error

18. The ASCII value of ‘A’ is 65. what is the value of y in the following program?

```
 main()
 {
  int y;
   y=‘B’- ‘A’/2;
   }
```

a. 0
b. Negative
c. 34
d. 1

19. short a[4][3]={ {1},{2,3),{4,5,6}};
 printf("%d\n", sizeof(a));
 Assuming a short is two bytes long, what will be printed by the above code?

a. compilation error
b. 6
c. 12
d. 24

20. Which escape character can be used to beep from speaker in C?

a. \a
b. \b
c. \n
d. None

UNIT III

```
1. int x = 5;
   int y = 2;
   char op = '*’;
   switch (op)
   {
     default : x +=1;
     case '+' : x += y; /*It will go to all the cases*/
    case '-' : x -= y;

    }
```

After the sample code above has been executed, what value will the variable x contain?

a. 4
b. 5
c. 6
d. 7
e. 8

2. What will be output if you will compile and execute the following c code?

```
 void main(){
 int i=10;
 static int x=i;
 if(x==i)
 printf("Equal");
 else if(x>i)
 printf("Greater than");
 else
printf("Less than");
}
```

a. Equal
b. Greater than
c. Less than
d. Compiler error
e. None of above

3. How is a variable accessed from another file?

a. The global variable is referenced via the extern specifier.
b. The global variable is referenced via the auto specifier.
c. The global variable is referenced via the global specifier.

d. The global variable is referenced via the pointer specifier.
e. The global variable is referenced via the ext specifier.

4. What will be the output?

```
main()
{
int a=1,b=1;
if(a==0)
if(b==0)
printf((“Hi”);
else
printf(“Bye”);
 }
```

a. Hi
b. Bye
c. HiBye
d. Nothing

5.What is the result of executing the following code?

```
int x=2;
if(x=1)  printf(“TRUE”);
else printf(“FALSE”);
```

a. TRUE
b. FALSE
c. unexpected result
d. compile-time error

6. What is the output of the following program?

```
main()
{
    int i =97;
    switch(i)
   {
     case ‘a’:
       if(i<13)
           case  ‘b’:i=10;i=11;
     printf(“%d\n”,i);
   }
 }
```

a. 97
b. 11
c. 10
d. Garbage Value

7. What will be the output of following code?

```
 main()
{
 int x=0;
 for(x=1;x<4;x++);
    printf("x=%d ",x);
 }
```

a. x=0
b. x=1
c. x=3
d. x=4

8. break statement is used to exit from

a. An if statement
b. A for loop
c. A program
d. The main() function

9. Which of the following code segments don't generate infinite loops:

a. int i, for(i=1;i;i--) {x=5;}
b. int i, for(i=1;i;) {x=5;}
c. int i=4; while(i){if(i==2) continue; x=5;i--}
d. for(; -4 ;){i=5;}

10. What will be the output of the following Program code:-

```
void main()
{
int x=0;
for(; ;)
{
 if(x++ = = 4)
 break;
 continue;
 }
 printf("x=%d",x);
```

a. x=0
b. x=4
c. x=5
d. loop will continue infinite times

11. What will be the output of the following Program code:-

```
 main()
{
 int i=0;
 while(i++<10)
{
 if(i<4 && i<9)
  continue;
  printf("%d ",i);
}
 }
```

a. 1 2 3 4 5 6 7 8 9
b. 1 2 3 10
c. 4 5 6 7 8 9 10
d. 4 5 6 7 8 9

12. What is the scope of a static variable?
a. Only in the program in which it is declared.
b. only in that function or block in which it is declared.
c. entire program plus other programs where variable is declared with extern
d. entire file plus other files where the variable is declared with extern.

13. How many times following loop will get executed?

```
main()
{
        int i=1;
        while(i<=10)
        {
          if(i==4))
          continue;
          printf("%d",i);
          i++;
        }
}
```

a. 10 times
b. 9 times
c. infinite times
d. 3 times

14. What will be the output of the following Program code:-

```
main()
{
        int i=,j,x=0;
        for(i=0;i<5;i++)
          for(j=i;j>0;j--)
               x=i+j+1;
               printf("x=%d",x);
}
```

a. 10
b. 9
c. 6
d. None

15. What will be the output of the following Program code:-

```
main ()
{
        int i=11;
```

```
            if(i=10)
            printf("BBDNITM");
            else
            printf("MCA");
        }
```

a. BBDNITM
b. MCA
c. Error
d. None

16. What will be the output of the following Program code:-

```
        main ()
        {
            int a=0,b=0;
            if(!a)
            {
              b=!a;
              if(b)
              a=!b;
            }
            printf("%d, %d\n", a,b);
        }
```

a. 1,0
b. 0,1
c. 1,1
d. 0,0

17. . what is the output of the following code?

```
# define SQ(a)   a*a
Main()
{
Int a=4,b;
B=SQ(a+1);
Printf("%d\n", b);
}
```

a. 6
b. 25
c. 9
d. 12
e. 36

18. int x=0;

```
 for (x=1;x<4;x++);
 printf("x=%d\n",x);
```

What will be printed when the sample code above is executed?

19. void myFunc (int x)

```
   {
    if(x>0)
    myFunc(--x);
    printf("%d, ", x);
}
```

```
int main()
{ myFunc(5);
  return(0);
}
```

a. 1, 2, 3, 4, 5, 5,
b. 4, 3, 2, 1, 0, 0,
c. 5, 4, 3, 2, 1, 0,
d. 0, 0, 1, 2, 3, 4,

20. int x=3;

```
    If( x==2);
    x=0;
  if( x==3)
x++;
else
x+=2;
else x += 2;
```

What value will x contain when the sample code above is executed?

a. 1
b. 2
c. 3
d. 4
e. 5

UNIT IV

1. What will be output if you will compile and execute the following c code?

```
void main(){
int i=320;
char *ptr=(char *)&i;
printf("%d",*ptr);
}
```

a) 320
b) 1
c) 64
d) Compiler error
e) None of above

2. What will be output if you will compile and execute the following c code?

```
void main(){
int array[]={10,20,30,40};
printf("%d",-2[array]);
}
```

a. 60

b- 30

c. 60

d. Garbage value

e. Compiler error

3. What will be output if you will compile and execute the following c code?

```
void main(){
int i=5,j=2;
if(++i>j++||i++>j++)
printf("%d",i+j);
}
```

a. 7

b. 11

c. 8

d. 9

e. Compiler error

4. What will be output if you will compile and execute the following c code?

```
void main(){
int array[3]={5};
int i;
for(i=0;i<=2;i++)
printf("%d ",array[i]);
}
```

a. 5 garbage garbage

b. 5 0 0

c. 5 null null

d. Compiler error

e. None of above

5. What will be output if you will compile and execute the following c code?

```
void main(){
int array[2][2][3]={0,1,2,3,4,5,6,7,8,9,10,11};
printf("%d",array[1][0][2]);
}
```

a. 4

b. 5

c. 6

d. 7

e. 8

6. What will be output if you will compile and execute the following c code?

```
void main(){
int a[2][4]={3,6,9,12,15,18,21,24};
printf("%d %d %d",*(a[1]+2),*(*(a+1)+2),2[1[a]]);
}
```

a.15 18 21

b.21 21 21

c.24 24 24

d. Compiler error

e. None of above

7. What will be output if you will compile and execute the following c code?

```
void main(){
int i;
double a=5.2;
char *ptr;
ptr=(char *)&a;
for(i=0;i<=7;i++)
printf("%d ",*ptr++);
}
```

a. -51 -52 -52 -52 -52 -52 20 64

b. 51 52 52 52 52 52 20 64

c. Eight garbage values.

d. Compiler error

e. None of these

8. What will be output if you will compile and execute the following c code?

```
void main(){
static char *s[3]={"math","phy","che"};
typedef char *( *ppp)[3];
static ppp p1=&s,p2=&s,p3=&s;
char * (*(*array[3]))[3]={&p1,&p2,&p3};
char * (*(*(*ptr)[3]))[3]=&array;
p2+=1;
p3+=2;
printf("%s",(***ptr[0])[2]);
}
```

a. math

b. phy

c. che

d. Compiler error

e. None of these

9. What will be output if you will compile and execute the following c code?

```
#include"conio.h"
int display();
int(*array[3])();
int(*(*ptr)[3])();
void main(){
array[0]=display;
array[1]=getch;
ptr=&array;
printf("%d",(**ptr)());
(*(*ptr+1))();
}
int display(){
int x=5;
return x++;
}
```

a.5

b. 6

c. 0

d. Compiler error

e. None of these

10. What will be output if you will compile and execute the following c code?

```
void main(){
int i;
float a=5.2;
char *ptr;
ptr=(char *)&a;
for(i=0;i<=3;i++)
printf("%d ",*ptr++);
}
```

a. 0 0 0 0

b. Garbage Garbage Garbage Garbage

c. 102 56 -80 32

d. 102 102 -90 64

e. Compiler error

11. char ** array [12][12][12];
Consider array, defined above. Which one of the following definitions and initializations of p is valid?

a. char ** (* p) [12][12] = array
b. char ***** p = array;
c. char * (* p) [12][12][12] = array;
d. const char ** p [12][12][12] = array;
e. char (** p) [12][12] = array;

12. struct customer *ptr = malloc(sizeof(struct customer));
Given the sample allocation for the pointer "ptr" found above, which one of the following statements is used to reallocate ptr to be an array of 10 elements?

a. ptr = realloc(ptr, 10 * sizeof(struct customer));
b. realloc(ptr, 9 * sizeof(struct customer));
c. ptr += malloc(9 * sizeof(struct customer));
d. ptr = realloc(ptr, 9 * sizeof(struct customer));
e. realloc(ptr, 10 * sizeof(struct customer);

13. Which one of the following is a true statement about pointers?

a. They are always 32-bit values.
b. For efficiency, pointer values are always stored in machine registers.
c. With the exception of generic pointers, similarly typed pointers may be subtracted from each other.
d. A pointer to one type may not be cast to a pointer to any other type.
e. With the exception of generic pointers, similarly typed pointers may be added to each other.

14. int testarray[3][2][2] = {1, 2, 3, 4, 5,
6, 7, 8, 9, 10, 11, 12};
What value does testarray[2][1][0] in the sample code above contain?
a. 3
b. 11
c. 9
d. 5

15. In C, a Union is
a. memory location
b. memory stored
c. memory screen
d. None of these

16. Maximum number of elements in the array declaration int a[5][8] is
a. 28
b. 32
c. 35
d. 40

17. char ** array [12][12][12];
Consider array, defined above. Which one of the following definitions and initializations of p is valid?
a. char ** (* p) [12][12] = array; [Ans]
b. char ***** p = array;
c. char * (* p) [12][12][12] = array;
d. const char ** p [12][12][12] = array;
e. char (** p) [12][12] = array;

18. char ptr[ ] "Hello World";
char *ptr2 = malloc(5);
ptr2=ptr1;
What is wrong with the above code (assuming the call to malloc does not fail)?
a. There will be a memory overwrite.
b. There will be a memory leak.
c..There will be a segmentation fault.
d. Not enough space is allocated by the malloc.
e. It will not compile.

19. int a [8] = { 0, 1, 2, 3 };
The definition of a above explicitly initializes its first four elements. Which one of the following describes how the compiler treats the remaining four elements?
a. Standard C defines this particular behavior as implementation-dependent. The compiler writer has the freedom to decide how the remaining elements will be handled.
b. The remaining elements are initialized to zero(0).[Ans]
c. It is illegal to initialize only a portion of the array. Either the entire array must be initialized, or no part of it may be initialized.
d. As with an enum, the compiler assigns values to the remaining elements by counting up from the last explicitly initialized element. The final four elements

will acquire the values 4, 5, 6, and 7, respectively.
e. They are left in an uninitialized state; their values cannot be relied upon.

20. Which one of the following statements allocates enough space to hold an array of 10 integers that are initialized to 0?
a. int *ptr = (int *) malloc(10, sizeof(int));
b. int *ptr = (int *) calloc(10, sizeof(int));
c. int *ptr = (int *) malloc(10*sizeof(int)); [Ans]
d. int *ptr = (int *) alloc(10*sizeof(int));
e. int *ptr = (int *) calloc(10*sizeof(int));

UNIT V

1. What function will read a specified number of elements from a file?
   a. fileread()
   b. getline()
   c. readfile()
   d. fread()
   e. gets(
2. f = fopen(filename, "r");
   Referring to the code above, what is the proper definition for the variable f?
   a. FILE f;
   b. FILE *f;
   c. int f;
   d. struct FILE f;
   e. char *f;
3. What is a proper method of opening a file for writing as binary file?
   a) FILE *f = fwrite("test.bin", "b");
   b) FILE *f = fopenb("test.bin", "w");
   c) FILE *f = fopen("test.bin", "wb");
   d) FILE *f = fwriteb("test.bin");
   e) FILE *f = fopen("test.bin", "bw");

4. If a variable has been declared with file scope, as above, can it safely be accessed globally from another file?
   a) Yes; it can be referenced through the register specifier.
   b) No; it would have to have been initially declared as a static variable.
   c) No; it would need to have been initially declared using the global keyword.

d) Yes; it can be referenced through the publish specifier.
e) Yes; it can be referenced through the extern specifier.

5. struct customer *ptr = malloc(sizeof(struct customer));
   Given the sample allocation for the pointer "ptr" found above, which one of the following statements is used to reallocate ptr to be an array of 10 elements?
   a) ptr = realloc(ptr, 10 * sizeof(struct customer)); [Ans]
   b) realloc(ptr, 9 * sizeof(struct customer));
   c) ptr += malloc(9 * sizeof(struct customer));
   d) ptr = realloc(ptr, 9 * sizeof(struct customer));
   e) realloc(ptr, 10 * sizeof(struct customer));

6. Which one of the following Standard C functions can be used to reset end-of-file and error conditions on an open stream?
   a) clearerr()
   b) fseek()
   c) ferror()
   d) feof()
   e) setvbuf()

7.
```
char ptr1[] = "Hello World“;
char *ptr2 = malloc( 5 );
ptr2 = ptr1;
```
   What is wrong with the above code (assuming the call to malloc does not fail)?
   a) There will be a memory overwrite.
   b) There will be a memory leak.
   c) There will be a segmentation fault.
   d) Not enough space is allocated by the malloc.
   e) It will not compile.

8. Which one of the following statements allocates enough space to hold an array of 10 integers that are initialized to 0?
   a) int *ptr = (int *) malloc(10, sizeof(int));
   b) int *ptr = (int *) calloc(10, sizeof(int));
   c) int *ptr = (int *) malloc(10*sizeof(int)); [Ans]
   d) int *ptr = (int *) alloc(10*sizeof(int));
   e) int *ptr = (int *) calloc(10*sizeof(int));

9. What are two predefined FILE pointers in C?
   a) stdout and stderr
   b) console and error
   c) stdout and stdio
   d) stdio and stderr
   e) errout and conout

10. How is a variable accessed from another file?
    a) The global variable is referenced via the extern specifier.[Ans]

b) The global variable is referenced via the auto specifier.
c) The global variable is referenced via the global specifier.
d) The global variable is referenced via the pointer specifier.
e) The global variable is referenced via the ext specifier.

11.
```
FILE *f = fopen( fileName, "r");
readData( f);
if( ????)
{ puts( "End of file was reached");}
```

Which one of the following can replace the ???? in the code above to determine if the end of a file has been reached?
a) f == EOF[Ans]
b) feof(f)
c) eof(f)
d) f == NULL
e) !f

12.
```
double read_double (FILE * fp){
  double d;
  assert(fp != NULL);
  fscanf(fp, " %lf", d);
  return d;}
```

13. The code above contains a common error. Which one of the following describes

it?
a) fscanf() will fail to match floating-point numbers not preceded by whitespace.
b) The format specifier %lf indicates that the corresponding argument should be long double rather than double.
c) The call to fscanf() requires a pointer as its last argument.
d) The format specifier %lf is recognized by fprintf() but not by fscanf().
e) d must be initialized prior to usage.

14. Which one of the following is valid for opening a read-only ASCII file? **a)** fileOpen (filenm, "r");
a. fileOpen (filenm, "ra");
b. fileOpen (filenm, "read");
c. fopen (filenm, "read");
d. fopen (filenm, "r");

15. f = fopen(filename, "r");
Referring to the code above, what is the proper definition for the variable f?
a. FILE f;
b. FILE *f;[Ans]
c. int f;

d. struct FILE f;
e. char *f;

16. If there is a need to see output as soon as possible, what function will force the output from the buffer into the output stream?
a) flush()
b) output()
c) fflush()
d) dump()
e) write().

17.

```
int debug (const char * fmt, ...){
  extern FILE * logfile;
  va_list args;
  assert(fmt);
  args = va_arg(fmt, va_list);
  return vfprintf(logfile, fmt, args);
  }
```

The function debug(), defined above, contains an error. Which one of the following describes it?
a) The ellipsis is a throwback from K&R C. In accordance with Standard C, the declaration of args should be moved into the parameter list, and the K&R C macro va_arg() should be deleted from the code.
b) vfprintf() does not conform to ISO 9899: 1990, and may not be portable.
c) Library routines that accept argument lists cause a fault on receipt of an empty list. The argument list must be validated with va_null() before invoking vfprintf().
d) The argument list args has been improperly initialized.
e) Variadic functions are discontinued by Standard C; they are legacy constructs from K&R C, and no longer compile under modern compilers.

18. Which one of the following calls will open the file test.txt for reading by fgetc?
a) fopen("test.txt", "r");
b) read("test.txt")
c) fileopen("test.txt", "r");
d) fread("test.txt")
e) freopen("test.txt")

19. Which one of the following will define a function that CANNOT be called from another source file?
a) void function() { ... }
b) extern void function() { ... }
c) const void function() { ... }
d) private void function() { ... }
e) static void function() { ... }

20. What function will read a specified number of elements from a file?
a) fileread()
b) getline()
c) readfile()
d) fread()
e) gets()

Short questions

Unit-I

1. List the basic rules which can be applied to TURBO Prolog variables?
2. Name two real life application of multithreading
. 3 What is generic subroutine. Name one language which support it?
4. What is the purpose of private part of an object interface?
5. Describe the difference between deep and shallow binding of reference ?
7. Discuss Structural and names equavalane for types. Give an example of language used for each approach?

8. What is thread? Explain atleast two options to create thread.
9. Give two difference between dynamic binding and polymorphism?
10. Explain
    a. P code
    b. Just in time Compiler

11. Differentiate between Procedural and Structured Language?
12. Write down the attribute of good programming language?

Unit-2

1. Write a program to interchange 2 variables without using the third one
2. What do the 'c' and 'v' in argc and argv stand for? Are the variables argc and argv are local to main?
3. Can we specify variable field width in a scanf() format string? If possible how?
4. What are register variables? What are the advantage of using register variables?
5. What do you mean by bitwise operators? Discuss them in detail
6. Write a C program to input any integer number by the user and test the bit value at a particular bit position.(staring from 0-15)
7. are unary operators? How many operands are associated with a unary operator?
8. Explain the following storage classes of C with example.
9. Resister storage class
10. External storage class
11. Static storage class
12. Can we specify variable field width in a scanf() format string? If possible how?

Unit-3

1. What is function pointer and where we will use it? What is the advantage of function pointer?
2. What is function pointer and where we will use it? What is the advantage of function pointer?
3. Can Structures be used with an Array?
4. What information does the header files contain?
5. What is the difference between NULL & NUL keywords in C?
6. Write code for initializing one dimentional and two dimentional array in a C Program?
7. What is the purpose of realloc( )?

8. What does static variable mean?
9. In header files whether functions are declared or defined?
10. What are the differences between malloc() and calloc()?
11. Difference between pass by reference and pass by value?
12. Explain the example, how we can replace for loop with while loop.

Unit-4

1. What will be output if you will compile and execute the following c code? And explain why?

```
void main(){
int array[2][2][3]={0,1,2,3,4,5,6,7,8,9,10,11};
printf("%d",array[1][0][2]);
}
```

2. What will be output if you will compile and execute the following c code with expalination?

```
void main(){
static char *s[3]={"math","phy","che"};
typedef char *( *ppp)[3];
static ppp p1=&s,p2=&s,p3=&s;
char * (*(*array[3]))[3]={&p1,&p2,&p3};
char * (*(*(*ptr)[3]))[3]=&array;
p2+=1;
p3+=2;
printf("%s",(***ptr[0])[2]);
}
```

3. When shoule be a far pointer is used?
4. What is function pointer and where we will use it? What is the advantage of function pointer?
5. Difference between null pointer and null macro?
6. Between a long pointer and char pointer which one uses more memory and why explain?
7. How do you insert an element into linear list without using pointer?
8. When would you use a pointer to a function?
9. What do you known by normalization of pointer?
10. Can Structures be used with an Array?
11. Difference between pass by reference and pass by value?
12. What is difference between Structure and Unions? What the advantages of using Unions.

Unit-5

1. What are the file mode in C language. Explain each?
2. How can you know given file is regular file, special file or regular file?
3. How can we know read/write permission any given file?
4. How can we know size and drive where file has stored of any given file?
5. Why we use file handling in C?
6. What is the difference between Binary search and Sequential search?
7. How do you read and write in Binary file in C language?
8. How to convert the content of file in upper case, file may contain both alphabetic and non alphabetic characters?
9. How delete a line in a file and replace it with another line.
10. What are the methods available in storing sequential file?
11. How is any Data Structure application is classified among files?
12. What is the bucket size, when the overlapping and collision occur at same time?

Long Questions

Unit-1

1. Define Syntax. What is general Syntactic Criteria? Explain its all points .
2. For a program variable what is scope? Differentiate b/w static scope & dynamic scope .what are the two main problem which can occur in static scoping?
3. What is programming Languages? Why there is need to study of Programming Languages? Explain its all reason with examples.

4. What is type checking of type checking? Define the Classifications of type checking. Mention their advantages and disadvantages.
5. a. mention Syntax and semantic for expressions in detail?
   b. Explain all type s of sub program environment.
6. Define sub program and its activation , how can you implement and activation record? Explain with diagram?

Unit-2

1. Discuss arithmetic, logical and relational operator with example and also give their order of precedence.
2. The increment(++) and decrement (--) operators can either be used as prefix or postfix. Will they always yield the same result?
3. Explain the usage of the conditional operators with the help of the suitable example?
4. Differentiate the following

   Unary, Binary and Ternary operator

5. What will be the output of the following C program? Justify your answer.

```
 #include<stdio.h>
#include<conio.h>
void main()
{
int a=6,b;
clrscr();
b=a+++--a;
printf("%d %d",b,a);
getch();
 }
```

6. What is Preprocessor? What are macros? what are its advantages and disadvantages?

Unit-3

1. Write a program that gives an arithmetic test to a person setting at the keyboard. The person responses to a question placed by the program and the program reads the answer. The program keeps on asking the user to try

again as long as the answer is wrong. Suitable messages are to be displayed during the runoff the program.

2. What criteria would you apply in deciding which of the three forms of to use in program.
3. What are the most efficient if-else ladder or switch statement?
4. Explain the functions getche() and putch () with the help of suitable examples?
5. Write short notes on
   b. Buffered I/O
   c. Putchar() function
   d. Call by value and Call by reference
6. Given the value of a positive integer N. Write a program segment which segment while determines how many decimal digits are needed to write down the value N (for example, 38 requires two digits, and 270 needs three digits)

Unit-4

1. What do you known by structure and self referential structure with suitable example?

2. A positive integer is called “Nasty” if has at least two pairs of positive integer factor such that the difference of one pair equals the sum of the other pair.

3. Write a program that uses self referential structures to create a linked list of nodes of the following structure. While creating the linked list, the student data is read. The list is also traveled to print the data.

4. How do you declare the following:
o An array of three pointers to chars
o An array of three char pointers
o A pointer to array of three chars
o A pointer to function which receives an int pointer and returns a float pointer
o A pointer to a function which receives nothing and returns nothing

5. Differentiate Between the following:
   a. Structure and Union
   b. Costant Pointer and Pointer to a Constant
   c. Array of Pointers and Pointer to an array
   d. malloc() and calloc() functions

6.Explain the meaning of following statements:

a. int *p[15];
b. int (*p)[15];
c. float *p(char *a[5]);
d. float*(*p[ ])(int *);
e. const int * const int p;.

Unit-5

1. Explain the meaning and syntax of following functions:-
   - fseek()
   - ftell()
   - fprintf()
   - fscanf()
   - fopen()

2. Write a function definition of Xstrcpy() and Xstrcat() in C ,which copies the contents of one string into the other and concatenates the one string after another respectively.

3.Write a C program to enter a list of names and sort them alphabetically in ascending order

4. What will be output of following program?

```
#include "stdio.h"

void main()

{

        char c;

        FILE *fp;

 fp=fopen("myfile.txt","w+");

 fprintf(fp,"you know");

 fclose(fp);

 fp=fopen("myfile.txt","r");

 clrscr();

 while((c=fgetc(fp))!=EOF)
```

```
        printf("%c",c);

    fclose(fp);

    getch();

    }
```

5. Write a program to read line-by-line from a given input file,, process each line (i.e. its words) and then move on to other line...

6. Do you mean by C Pre-Processors? Explin their functioning. Also Explain the various types of preprocessor direcives with suitable examples