Fardouse Lomat Jahan Rumpa
ID : 2219170041
Department : CSE (E)
Batch : 17th

Course Code : CSE-121

Course name : Structured Programming Language

CSI-121 (Programming Language) Structure

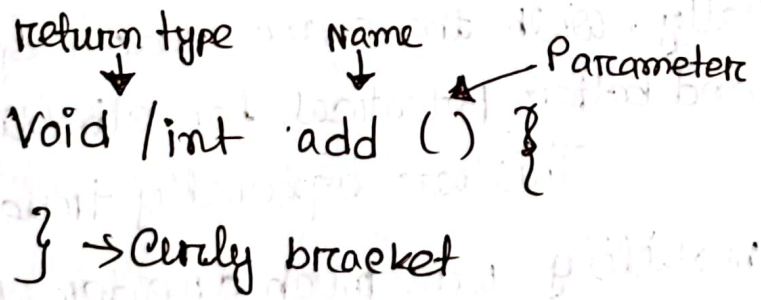
Ans to the Qus No: 01

(a) Answer: **Function Declarations:**

A function declaration tells the compiler about a functions name, return type, and parameters. A function definition provides the actual body of the function.

```
Void /int fun ( ) {
Print f (" hello world ");
}
```

(b) Answer: In order to declare function, ~~first~~ firstly we have to declare the return type then the name of the function after words we need to put parameters and finally a pair of curly brackets. See the following example.



(c) Answer: The return type will be int and the function will receive one fractional or float number and another one will integeer number.

Ans to the Qus No: 02

(a) Answer: An array is a collection of similar data elements sorted at contiguous memory locations.

Declaration: First of all we need to declare the type of array then the name of the array and then we have to put the number how many elements it can be sorted. for an example:

```
int  list  [5];
 ↓    ↓
type  Name
```

array initializations:

```
int list [5] = { 1,2,3,4,5};
```

(b) Answer:

a[2][0] values is 30

a[2][1] Values is 40

a[1][1] Values is 24

(c) Answer: Two Dimensional array:

The two-dimensional array can be defined as an array of arrays. The 2D array is organized as Matrices which can be represented as the collection of

rows and columns. However, 2D arrays are created to implement a relational database lookalike data structure. It provides ease of holding the bulk of data at once which can be passed to any number of functions whenever required.

```
int X [5] [5];
```

Multi-dimensional Arrays:

A multi-dimensional array can be termed as an array of arrays that stores homogeneous data in tabular form. Data in multi-dimensional are stored in row-major order.

```
int X [5] [5] [5];
```

(d) Answer:

Null Pointer: At a very high level we can think of Null ass a null pointer which is used in C for various purposes. ~~Some of the~~

```
int * pInt = NULL;
```

Ans to the Qus No: 03

(a) Ans: **Structure:**

A structure can be defined as a single entity holding variable of different data types that are logically related to each other. All the data members inside a structure are accessible to the functions defined outside the structure. To access the data members in the main function, you need to create a structure variable.

Syntax to Define a Structure in C

```
Struct structName
{
   // structure definition
Data_type1 member_name1;
Data_type2 member_name2;
Data_type2 member_name2;
};
```

Example:

```
Struct My structure { // structure
declaration
   int MyNum;          // member
(int variable)
    char myLetter;     // member
(Char variable)
}; // End the structure with a semicolon.
```

(b) Answer: Steps to Read a file:

→ Open a file using the function fopen() and store the reference of the file in a FILE pointer.

→ Read contents of the file using any of these functions fgetc(), fgets(), fscanf() or fread().

fgetc()

```
int fgetc(FILE *ptr);
```

Whole line from text file:

```
#include <stdio.h>
Void main() {
int lineNum=0;
char lineContent[100];
Scanf("%s", &lineContent);
Preintf("%S\n", Line content);
}
```

(c) Answer: File:

A file is a collection of data stored in one unit, identified by a filename. It can be a document, picture, audio or video stream, data library, application, or other collection of data.

The fopen() function is used to create a file or open an existing file:

```
fp= fopen (const char filename , const char
    mode);
```

To close a file, use the member function close().

```
xyst.close();
```

The close function takes no parameters and returns no values.

(d) Answer:

File handling is one of the most important topics in the C language. In case of file handling, file opening modes play an important role in executing these programs. There are several types of modes available in the file opening mode. Each of them will discuss in our article.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{ int n;
 file *fp;
 fp = fopen ("Write.txt", "w"); //
  file opening mode.
  if (fp == NULL)
 {printf ("File not found!!");
  exit (1);
```

```
}
n = 2351;
fprintf(fp, "%d", n); // Usages of fprintf() function
Fclose(fp); // file is closed.
return 0;
}
```

Ans to the Qus No: 05

(a) A pointer is supremely useful in C programming. They are extremely efficient to use which is what makes them popular. We need to declare pointers to access that particular address or to get the reference to the variable declared. Pointers are often used to get a faster execution time.

Syntax:

datatype * pointer_variable name;

Example: int * ptr1;

Explanation:

For pointer declaration in C, you much make sure that the data type you're using is a valid C. data type and that the pointer and the variable to which the pointer variable points must have the same data type.

For example, if you want a pointer to point to a variable of data type int, i.e. int var = 5 then the pointer must also be of data type 'int', i.e. int *ptr = &var. The * symbol indicates that the variable is a pointer. To declare a variable as a pointer, you're

must prefix id with &.

In the example above, we have done a pointer declaration and named ptr1 with the data type Integer.

(b) Ans:

(i) strcpy(): strcpy() function is used to copy a character variable.

Syntax:

```
Char city [15]
strcpy (city, "Bangladesh"]
```

This will assign the string "BANGLADESH" to the character variable city.

Note: That character value like city = "BANGLADESH"s cannot assign in c language.

(ii) Strlen() function:

Strlen() function is used to find the length of a character string.

Syntax:

```
int n;
char st [20] = "BANGLADESH";
n = strlen (st);
```

This will return length of the string 10 which is assigned to an integer variable n.

Note: That the null character '\0' available at the end of a string is not counted.

(iii) Strstr(): The strstr() function returns pointer to the first occurrence of the matched string in the given string. It is used to return substring

from match till the last characters.

Syntax:

char * strstr (const char * string, const char * match)

(c) Answer:

258^44 = 3910