Assignment - 3

PAGE No.
DATE / /

1] What are the different standardization of c programming?

→ **C programming Language**

1] C programming language is the procedural programming language.

2] C programming language was created by Dennis ritchi and kem's thomson in ATNT Bell lab.

3] initially c programming language considered as system programming language but later on it gets considered as general purpose programming language.

4] C programming language was standarized programming language.

standarization of C language

1) C programming language was standarized by multiple organization.

2) following are the standards of the C programming language.

1) K & R Standard

- Kernighan and Ritchie (K&R) standard the first edition of the C programming language book by Brian Kernighan and Dennis Ritchie was published in 1978.

- The book was one of the most successful computer science books and has Served as an informal standard for the C language for many years this informal standard was known as 'K&R' standard in C.

PAGE No. DATE / /

2] ANSI standard:

- ANSI stands for American national standard institute

- A technical committee was created under the American national standards institute (ANSI) committee to establish a standard specification of C in 1989.

- The standard proposed by the committee was formally approved and is often referred to as ANSI C

3] ISO standard

- In 1990, the international organization for standardization (ISO) adopted the ANSI C standard after minor modification

- This version of the standard is called ISO standard in C or this sometimes in C89.

3] According to the above standarization C language get standarized by multiple standarization team.

PAGE No.
DATE / /

2] What is meant by Data structure?

→ Data structure:-

- Data structure is defined as a way of storing and representing the data in particular format

- By using the datastructure we can store and represent the data in particular format

- The concept of Data struce is language independant

- Data structure is applicable for all programming language like C, C++, Java

- There are two types of data structure.

1) linear Datastructure

2] Non- linear Datastructure

PAGE No.
DATE / /

- Data structure provides systematic mechanism for storage, reterival and manipulation of data
for example:-
Arrays, stacks, queue linked lists, trees etc.

3] What is Array. explain in detail?

→ Array:-

1] Array is a collection Data structure which is come in linear Data structure type.

2] Array is used for the storage of homogeneous data i.e data of the same type

3] Array is considered as derived datatype in C, C++ and Java

For example:-

int Arr [5] = {10, 20, 30, 40, 50}

	0	1	2	3	4
Arr	10	20	30	40	50

- Here we create the Array of integer datatype i.e. int that data elements of that array are store sequentially i.e: one by one

Syntax for creating Array

Datatype ArrayName [Array length] = { Data element };

↑ Subscript operator

- In C, C++, Java the Array index always start from 0.

- when we Allocate the memory for Array all the elements ~~are~~ in array gets sequence memory

- Each elements from Array has its unique memory index

PAGE No.
DATE / /

- In case of Array we can Access all the elements using single name so there is no need to multiple names.

- int Arr [5] = { 10, 20, 30, 40, 50 }

- we can read the Array as follows.

- Array is one dimensional Array which contains 5 elements in it.

- Each element in it has integer datatype elements initalize 10, 20, 30, 40, 50.

Array is classified as

1] Single - dimensional arrays

2] Multi - dimensional arrays.

DATE / /

4] Differentiate the constant and variable

→ Variable:-

- A variable is an entity whose value can vary (i.e change) during the execution of a program.

- The value of variable can be changed because it has a modifiable l-value
- l-value can be placed on the left side of the assignment operator

- The variable can also placed on the right side of the assignment operator

- when we create a variable its considered as data object

- Each ~~object~~ data object contains two things has L-value & R-value

Syntax to create Variable

Datatype name-of variable = value;

Constant

- Constant is an entity whose value remains the same throughout the execution of a program

- It cannot be placed on the left side of the assignment operator. because it does not have a modifiable l-valu

∴ It can only be placed on the right side of the assignment operator thus, a constant has an r-value only
constants are classified are

1] Literal constants

2] Qualified constants

3] Symbolic constants.

DATE / /

5] Difference between local and Global variable.

Local variable	Global variable
1] Local variable create inside function body	1] Global variable is created outside the function body.
2] local variable is variable which is Accessable only in function body	2] global variable are accessable to all function which are defined in that program.
3] local variable is required in the program	3] Global variable Not required for global variable.
4] More reliable and secure	4] Accessible by multiple functions.
5] local variable value cannot be changed by other functions	5] Global variable values can be changed.

PAGE No. DATE / /

6] Explain the concept of function, explain about the starting point function in C, C++, java

→ Function

- Every programming language C, C++ and java provides the concept of function.

- function is defined as a named block

- By using the function we achieve reusability

- The body of a function consists of a set of statement enclosed within curly brackets commonly know as brackets.

Syntax of Declaration of function

```
returntype    functionname()
  {
     .... Business logic....
  }
```

DATE / /

- when we define the function that function can Accepts the parameters from users and will return the results as a return value.

- Every c, c++ and java application will start their execution from main () function

- The code written inside the main() function body gets executed

7] Explain about the programming language Paradigms in c, c++, Java

→

Paradigms

- A programming paradigms is a model of programming based on distinct concepts that shapes the way programmers design, organize and write program.

PAGE No.
DATE / /

- In case of any programming language we have to consider major 3 paradigms
  1) sequence
  2) selection
  3) Iteration

1) Sequence

- If the program contains multiple statement and if we execute that statement in a sequential manner then we consider as a sequence

- In the sequence paradigms the program Executing the statements in step by step.

- There is no specific syntax for the sequence paradigms

2) selection

- There are multiple options (choices) then we select the desired option Base on the requirement then it is considered as a selection.

DATE

- we select execution paths based on conditions.

- followings are comes under the ~~sequence~~ ~~pa~~ selection paradigms

if, if-else, switch.

3] Iteration.

- If we want to execute one or more than one statement multiple times then we use the concept of iteration

- Execute 1 or more than 1 statement multiple times then we use iteration paradigms.

- followings are comes under iteration paradigms.

for, while, do-while loop

PAGE No.
DATE / /

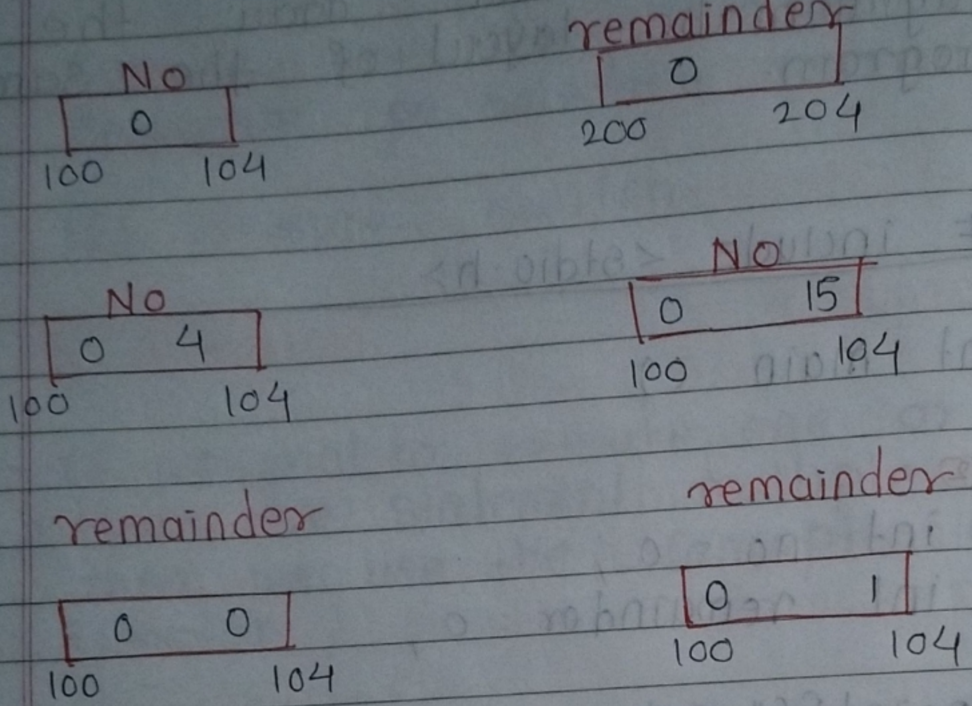
8] Write a program to accept a input from user and find whether the no is even or odd. draw the diagrammatic layout of the same program.

```
# include <stdio.h>

① int main ()
{
   ②
   int no = 0;
   int remainder = 0;

   printf(" Enter number : \n");
   scanf(" %d", & no);                ③

   remainder = no%2;                  ④

   if (remainder = = 0)
   {
      printf (" number is even");
   ⑤ }
   else
   {
      printf(" numeber is odd ");     ⑥
   }
   return 0;
}                                     ⑦
```

sequence

selection

Diagramatic layout of the program



0 = = 0 => true

1 = = 0 ⇒ false

number is even

number is odd

PAGE No.
DATE / /

9] Write notes on short hand operator

Short hand operator

In C, C++, Java there are two operators which are used to incremed (++) the value of 1 as well as used to decrement (--) & the value decreased by 1

- following are the short hand operators.

1] Increment (++) operator

2] Decrement (--) operator.

1] Increment (++) Operator

- The increment operator can be increm ent by 1.

- there are two types of increment operator.

1] Pre- increment operator

2] Post - increment operator.

DATE / / /

1) Pre- increment operator

- when the increment operator appears towards the left side of the operand it is known as pre-increment operator

- for example:-
No = ++i;

2) Post-increment operator

- when the increment operator appears towards the right side of its operand it is known as post-increment operator

for example:-
No = i++;

2] Decrement (--) operator

- In the Decrement operator value can be decrese by 1

- There are two types of Decrement operator

1) Pre-Decrement operator
2) Post- Decrement operator.

1) Pre- Decrement operator

- In the pre Decrement operator first we Decrease the value by 1 and the assign the values for operators

for example:- No = --i;

2) Post - Decrement operator

- In the Post - Decrement operator first we Assign the value then the value decrement by 1

for example:- No = i--;

PAGE No.
DATE / /

10] Write down notes on various loops with its syntax and example along with the dry run diagram

→ - when we execute 1 or more than 1 statement multiple times then we use the concept of iteration

- iteration is one of the programming language paradigms in C, C++, Java

- under the iteration pardigms following loops are comes.

1] for loop

2] while loop

3] do-while loop

1] For loop

- It is generally use when we already known the no. of iterations that we want to execute then we use the for loop.

For example:-

```
# include <Stdio.h>
OS → int main()
{
        int i = 0;
            ①                  ②          ③
for (i=1 ;       i<=5;      i++)
{
    printf("%d jay ganesha...\n", i); ← ④
}

return 0;
}
```

① → initialize loop counter

② → conditions

③ → increment / decrement of loop counter

④ → loop body

Dry Run:-

iteration 1:
1: i=1
2: 1<=5 → true
4: jay ganesha...
3: i++

i: ~~0~~ 1

iteration 2: i = 2
2: 2<=5 → true
4: jay ganesha...
3: i++

i: ~~1~~ 2

i: ~~2~~ 3

i: ~~3~~ 4

iteration 3 : i=3
2: 3<=5 → true
4: jay ganesha...
3: i++

i: ~~4~~ 5

iteration 4 : i=4
2: 4<=5 → true
4: jay ganesha...
3: i++

i: ~~5~~ 6

iteration 5 : i = 5
2: 5<=5 → true
4: jay ganesha
3: i++

iteration 6 : i = 6
2: 6<=5 → false
break

2) while loop

- It is generally use when we know the condition till we want to execute the code then we use while loop.

For example:-

```
# include <stdio.h>
int main ()
{
① int i = 1;

   while (i<=5) ②
   {
      printf ("%d Jay ganesha...\n", i); ④

      i++ ; ③
   }

 return 0;
}
```

Dry run

Step 1: i = 1 — i $\boxed{1}$

iteration 1 — i $\boxed{\cancel{1}\ 2}$

Step 2: 1 <= 5 → true
Step 4: 1. Jay ganesha...
Step 3: i++

DATE / /

iteration2 step 2: 2<=5 → true
step 4: 2. Jay ganesha...
step 3: i++

i
| ~~2~~ 3 |

iteration 3 step2: 3<=5 → true
step 4: 3. Jay ganesha...
step 3: i++

i
| ~~3~~ 4 |

iteration 4 step 2: 4<=5 → true
step4: 4. Jay ganesha...
step3: i++

i
| ~~4~~ 5 |

iteration5 step2: 5<=5 → true
step4: 5. Jay ganesha...
step 3: i++

i
| ~~5~~ 6 |

iteration 6 step 2: 6<=5 → false
break

3) Do-while

Do-while loop same as while loop but its executed atleast one respective of the condition

PAGE No.
DATE / /

For example:-

```
#include <stdio.h>
int main()
{
    int i = 1;
    do
    {
        printf("%d jay ganesha...\n", i);
        i++;
    }
    while (i<=5);

return 0;
}
```

A

B

Dry run:-

Step 1 : i = 1

i: 1

iteration 1:
// without checking condition
step 4: 1. jay ganesha...

i: ~~1~~ 2

iteration 2: Step2: 2<=5 → true
Step 4: 2. jay ganesha.
Step 3: i++;

i: ~~2~~ 3

iteration 3: step2: $3<=5 \rightarrow$ true
step4: 3. Jay ganesha..
step3: i++;

i
| ~~3~~ | 4 |

iteration 4: step2: $4<=5 \rightarrow$ true
step4: 4. Jay ganesha...
step3: i++;

i
| ~~4~~ | 5 |

iteration 5: step2: $5<=5 \rightarrow$ true
step4: 5. Jay ganesha..
step3: i++;

i
| ~~5~~ | 6 |

iteration 6: step2: $6<=5 \rightarrow$ false
break