Assignment - 12

Q1] What is meant by reference in C++?

Ans :) Reference is considered as a derived data type in C++ as well as Java.

:) When we create reference to an existing variable it is just considered as another name to that variable.

:) As it is just an another name, there is no ~~sp~~ seperate memory allocation for that variable.

:) To create a reference we use '&' operator

:) If the '&' is used after assignment operator then it is considered as address of operator

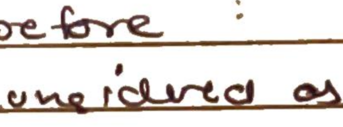
:) If the '&' operator is used before assignment operator then it is considered as reference operator

```
int no = 11;
int &ref = no;
```

:) no is variable of type integer, initialized with 11

:) ref is reference which refers to integers, currently it refers to variable no.

:) ref is ~~nam~~ another name of an integer & the name of original variable is no.



:) When we create a reference there is no seperate memory allocation for it

:) The name of reference & the original name refers to the same memory location.

:) Due to which address is same & value is same.

:) If we change the value of a variable using its original name, then value of reference variable gets changed automatically

:) When we create a reference its entry gets added inside symbol Table ~~inside~~

:) Symbol Table contains one column named as "Another name" which contains name of that ~~varia~~ reference variable.

name	Address	SIZE	value	Datatype	Another name
no	100	4	11	int	ref

:) Reference concept is used in
1) copy constructor
2) call by reference.

Q I] What is difference between pointer & reference?

Ans In pointers, the '&' ~~ope~~ operator is used after the assignment operator to store the address of the variable.

In reference, the '&' operator is used before assignment operator, it ~~dt~~ is just another name for the variable existing.

~~While pointer stores address of variable~~

Q5] Draw symbol table for below syntaxe.

```
int no = 11;     // consider address of no as 100
int *p = &no;    // consider address of p as 200
int **q = &p;    // consider address of q as 300
```

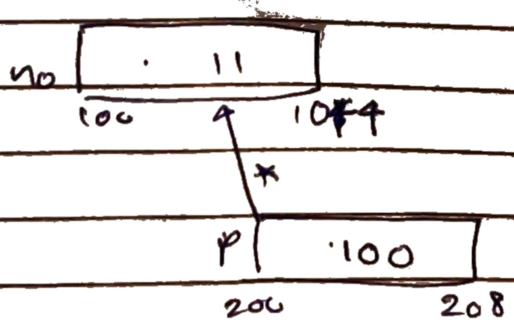
Name	Address	Size	Value	Datatype	Anotherm
no	100	4	11	int	-
p	200	8	100	pointer	-
q	300	8	~~p~~200	pointer	-

Q6] Draw symbol table & diagrammatic layout

```
int no = 11;      // Address 100 of
int *p = &no;     // consider address of p as 2
int *q = &p;      // consider address of q as 3
```

*

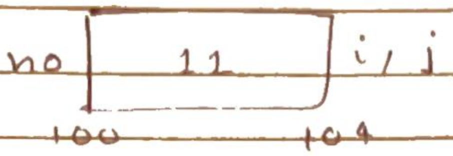
Name	Address	Size	Value	Datatype	Anothe
no	100	4	11	int	-
p	200	8	100	pointer	-



Q.7] Draw symbol table & diagrammatic representation

```
int no = 11 ;      // Consider Address of no as 100
int &i = no;
int &j = no;
```

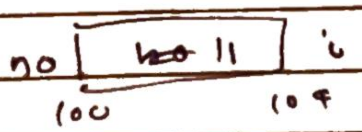
Name	Address	Size	Value	Datatype	Another Name
no	100	4	11	int	i, j

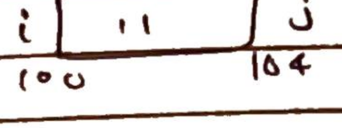


Q8] Draw symbol table & diagramatic representation

```
int no = 11;     // Consider address of no as 100
int &i = no;
int &j = i;
```

Name	Address	Size	Value	Datatype	Another Name
no	100	4	11	int	i
i	100	4	11	int	j





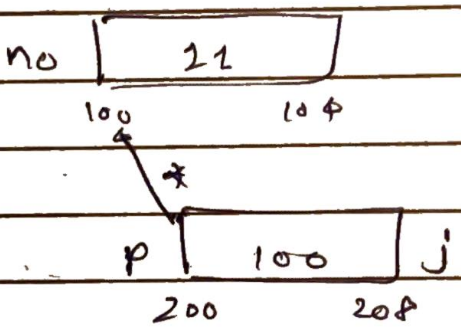
PAGE No.

DATE

Q10] ~~[illegible]~~ Draw symbol table & diagrammatic representation of below syntax

```
int   no = 11;      // consider address of no as 100
int * p  = &no;     // consider address of p as 20
int *(&j) = p;
```

Name	Address	Size	Value	Datatype	Another Name
no	100	4	11	int	-
p	200	8	100	pointer	j



no
11
100
104
*
p
100
j
200
208