# Computer Science
## Class XII
## POINTERS

Q1 EXPLAIN THE STATEMENT int * ptr ;

Q2 WHAT IS THE DIFFERENCE BETWEEN STACK AND HEAP?

Q3 MAKE A DIAGRAM OF CONCEPTUAL MEMORY MAP.

Q4 WHAT IS THE DIFFERENCE BETWEEN A POINTER VARIABLE AND A SIMPLE VARIABLE?

Q5 WHAT IS FREE POOL OF MEMORY?

Q6 GIVE AN EXAMPLE OF DECLARATION AND INITIALIZATION OF POINTERS

Q7 WHAT IS NULL VALUE ?

Q8 WHAT ARITHMETIC OPERATIONS ARE POSSIBLE ON POINTERS? HOW ?

Q9 EXPLAIN new AND delete OPERATORS

Q10 WHAT IS THE RETURN VALUE OF new WHEN THERE IS INSUFFICIENT MEMORY AVAILABLE ?

Q11 WHAT IS THE LIFETIME OF A VARIABLE CREATED WITH new?

Q12 DOES delete OPERATOR ACTUALLY DELETES THE MEMORY ?

Q13 WHAT IS MEMORY LEAK ? WHAT ARE ITS CAUSES ?

Q14 DIFFERENTIATE BETWEEN POINTER TO CONSTANT AND CONSTANT POINTER.

Q15 WHAT IS THE DIFFERENCE BETWEEN PASS BY REFERENCE AND PASS BY POINTER ?

Q16 IS (*P)++ AND ++*P SAME ?

Q17 ARE THE ABOVE STATEMENTS VALID?
int *p=60;
char *p=”hello”;

Q18 EXPLAIN INT *P[10];

Q19
```
int * p[5];
int a=1,b=2,c=3,d=4,e=5;
p[0]=&e;  p[1]=&d;  p[2]=&c;  p[3]=&b;  p[4]=&a;
cout<<p; cout<<*p; cout<<*(*p); cout<<**(p+3); cout<<**p+5;
        cout<<**(p+1) + 7;
```

Q20
```
int x[3][3]={ { 3,2,1},{6,5,4},{9,8,7}};
int *p=&x[0][0];
cout<<(*(p+1)+1);
cout<<*(p+2);
cout<<*(*p+2)+6;
cout<<++*p;
cout<<*(*(p+2) +1);
```

Q21 WHAT IS THE ADVANTAGE OF ARRAY OF POINTERS TO STRINGS OVER ARRAY OF STRINGS? .

Q22 WHAT ARE SELF REFERENTIAL STRUCTURES?

Q23 WHAT IS this POINTER? WHY IT IS USED?

Q24 EXPLAIN char **ptr? IF ptr=&p; AND p=&a AND char a='G';
WHAT IS THE O/P O F THE FOLLOWING STATEMENTS?
```
cout<<a;
cout<<*p;
cout<<*ptr;
cout<<**ptr;
cout<<ptr;
cout>>p;
```

Q25 WRITE A C++ PROGRAM THAT TAKES A STRING FROM THE USER AND USING POINTERS DO THE FOLLOWING-------

A. CHECK IT'S A PALINDROME OR NOT
B. COUNT NO. OF VOWELS
C. COUNT NO. OF WORDS
D. CONVERT ALL E'S TO I'S
E. REVERSE EACH WORD.

**NOTE:** USE DYNAMIC INITIALIZATION OF THE STRING

Q26 DIFFERENTIATE BETWEEN STATIC AND DYNAMIC ALLOCATION OF MEMORY .