Ques1. Define primary and derived data types with example.

Ans. • Primary data type (built-in) –

1. Integer
2. Float
3. Char
4. Double

• Derived data types

1. Arrays
2. Functions
3. Pointers

Ques(2) Write a C++ program to print the first and last digit of a given number using do while loop.

Ans.

```
#include<iostream.h>
#include<conio.h>
void main()
{
int lastdigit, firstdigit, no, copyno;
cout<<"Enter any num:";
cin>>no;
copyno=no;   // Gets last digit
lastdigit = no%10;
while (copyno>=10)
{
 copyno = copyno/10;
}
firstdigit = copyno;
```

```
cout<<"\n lastdigit"<< lastdigit;
cout<<"\n firstdigit"<< firstdigit;
getch();
}
```

Q2. Explain operator overloading mechanism to find the sum of 3 numbers.

2016

Q. Explain the terms: Class, exception handling, Call by value.

Ans. Call by value :- This approach is also popularly known as passing arguments by value. In this, approach the values of the actual arguments are passed to the function during function call. When control is transfer-red to the called function, the values of actual arguments are copied to the corresponding formal arguments and then, the body of the function is executed. It is the called function supposed to return a value, it is returned with the help of return statement.

Ques Name any three library functions and any three preprocessor directives in C++.

Ans - Preprocessor directives :-

(a) Inclusion directives - This category has only one directive which is called # include. This inclusion directive is used to include files into the current file.

(b) Macro definition directives - These are used to define macros which are one or more program statements like functions and they are expended in ~~time~~ line. They

include #define & #undef.

(c) Conditional compilation directives — These are used to execute statements conditionally for debugging purposes, executing the code on different machine architectures etc. These this include if, #elif, #ifdef and #ifndef.

Quss Why do we need different access specifier in class?

As. Access specifiers in C++ class defines the access control rules. They are used to set boundaries for availability of members of class be if data members of member functions. They set the accessibility of classes, methods and other members. They are a specific part of programming lang.

Ques- W.A.P. in C++ to calculate and display area A and parameter P of a rectangle R using classes. Given that rectangle R of length l and breadth b, area = l*b and parameter P = a+(l+b).

A -
```
#include <iostream.h>
#include <math.h>
class rectangle
{
int length;
int breadth;
```

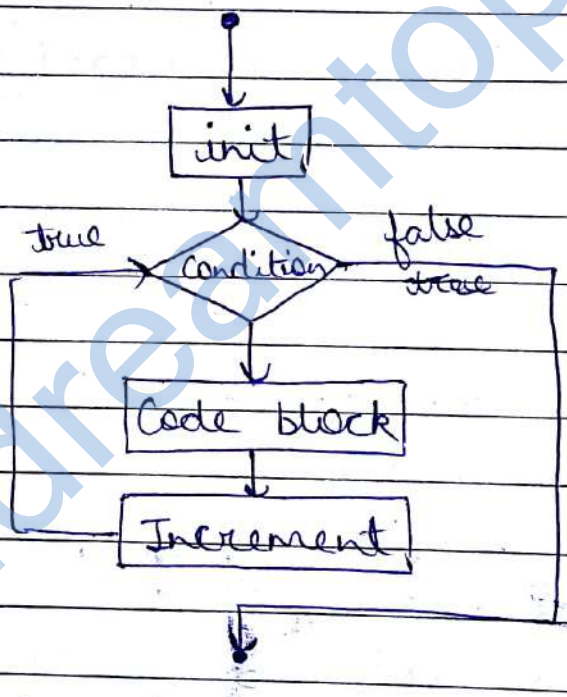
```
public :
  int getarea (int l, int b)
  {
  int area ;
  area = l * b ;
  cout << " Area :" << area << "\n" ;
  }
int getperameter (int i , int b)
  {
  int pert;
  pert =  + (l + b);
  cout << " Parameter " << part << "\n";
  }
  };
int main()
{
int l, b;
rectangle R ;
cout << " Enter l of rect ";
cin >> l ;
cout << " Enter b of rect ";
cin >> b ;
R. getarea ();
R. getparameter ();
return 0;
}
```

Ques- Explain for loop in C++.

Ans- A for loop is a repetition control structure that allows us to efficiently write a loop that needs to execute a specific number of times.

Syntax:-

```
for (init. ; condition ; increment)
{
statement( );
}
```



Ex-
```
# include <iostream.h>
int main()
{
for ( int a = 10; a < 20; a = a+1 )
{
cout << " Value of a :" << a << endl;
}
return 0;
}
```

Output – Value of 'a' : 10
" " " 11
" " 12
" " 19

Que Two single-D arrays A and B contain the elements as follows–
A[9] = 2,4,8,32,16,60,70,89,98
B[6] = 3,7,9,30,35,24
W.A.P that merges A and B and gives a third array C as follows: C[15] = 2,3,4,7,8...98

Ans.
```
# include <iostream.h>
# include <conio.h>
void main()
{
   int A[9], B[6], size1, size2, size, i, j, k, c[15];
   cout << "Enter array 1 size:";
   cin >> size1;
   cout << "Enter array 1 elements:";
   for (i=0; i<size1; i++)
   {
      cin >> A[i];
   }
   cout << "Enter array 2 size:";
   cin >> size2;
   cout << "Enter array 2 elements!";
   for (i=0; i<size2; i++)
   {
      cin >> B[i];
   }
```

```
for(i=0; i<size1; i++)
{
c[i] = A[i];
}
size = size1 + size2;
for(i=0; k=size1; k<size<
{
c[k]<B[i];
}
cout<<"Now the new array after merging:"
for(i=0; i<size; i++)
{
cout<<C[i]<<" ";
}
getch();
```

Ques - Discuss formatted and unformatted I/O opⁿs in stream classes.

Ans. C++ provides both the formatted and unformatted IO functions. In formatted or high-level IO, bytes are grouped and converted to types such as int, double, string or user-defined types. In unformatted or low-level IO, bytes are treated as raw bytes and unconverted. Formatted IO operations are supported via overloading the stream insertion (<<) and stream extraction (>>) operators, which presents a consistent public IO interface

To perform input and output, a C++ program:

1- Constructs a stream object.
2- Connects the stream object to an actual IO device (e.g. keyboard, console, file network, another program).
3- Performs input/output operations on the stream, via the functions defined in the stream's public interface in a device independent manner.
4- Disconnects the stream to the actual IO device (ex- close the file).
5- Frees the stream object.

C++ IO is provided in headers <iostream>
The <iostream> header declares these standard stream objects.

(i) cin (of istream class), wcin (of iostream class)
(ii) cout (————), wcout (" " ")
(iii) cerr (————), werr (————) standard error stream
(iv) clog (————), wclog (————), corresponding to the standard log stream, defaulted to display console.

Q Manipulators :-
Manipulators are helper functions that make it possible to control input/output stream using operator << or

operator >>. The manipulators that are invoked without arguments (ex. std::cout<< std::boolalpha; or std::cin >> std::hex;) are implemented as functions that takes a reference to a stream as their only argument.

Manipulators are stream functions available in "iomanip.h" headerfile and are used to change the default formats of input and output. These are used with stream insertion and estraction operators. They are are used for defining a specified format of input and output. They provide features similar to that of IOS member functions.

Some of the standard manipulators available in iomanip.h headerfile are endl, setw, setfill, hex, oct, dec, setprecision, flush, setiosflags etc.

Manipulator are different from ios member functions as manipulator does not return the previous format state as is the case with ios member functions.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int i,j,m,n,a[8][8], sum=0;
printf("Enter the order of the matrix:");
scanf("%d%d",&m,&n);
if(m==n)
{
printf("Enter elements in the matrix:%d",m*n);
for(i=0; i<m; i++)
{
for(j=0;j<n;j++)
}
printf("The enter matrix is:\n",m,n);
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
printf("%4d",a[i][j]);
if(i==j)
sum = sum + a[i][j];
}
printf("\n");
}
printf("Sum of diagonal elements of matrix is:%d\n",sum);
}
else
printf("Matrix should be a square matrix");
getch();
}
```

Ques. How delete[] is different from delete?

Ans. 1- delete is an operator whereas delete() is a library function.

2- delete free the allocated memory and calls destructor. But delete() de-allocate memory but does not call destructor.

3- delete is faster than delete() because an operator is always faster than a function.