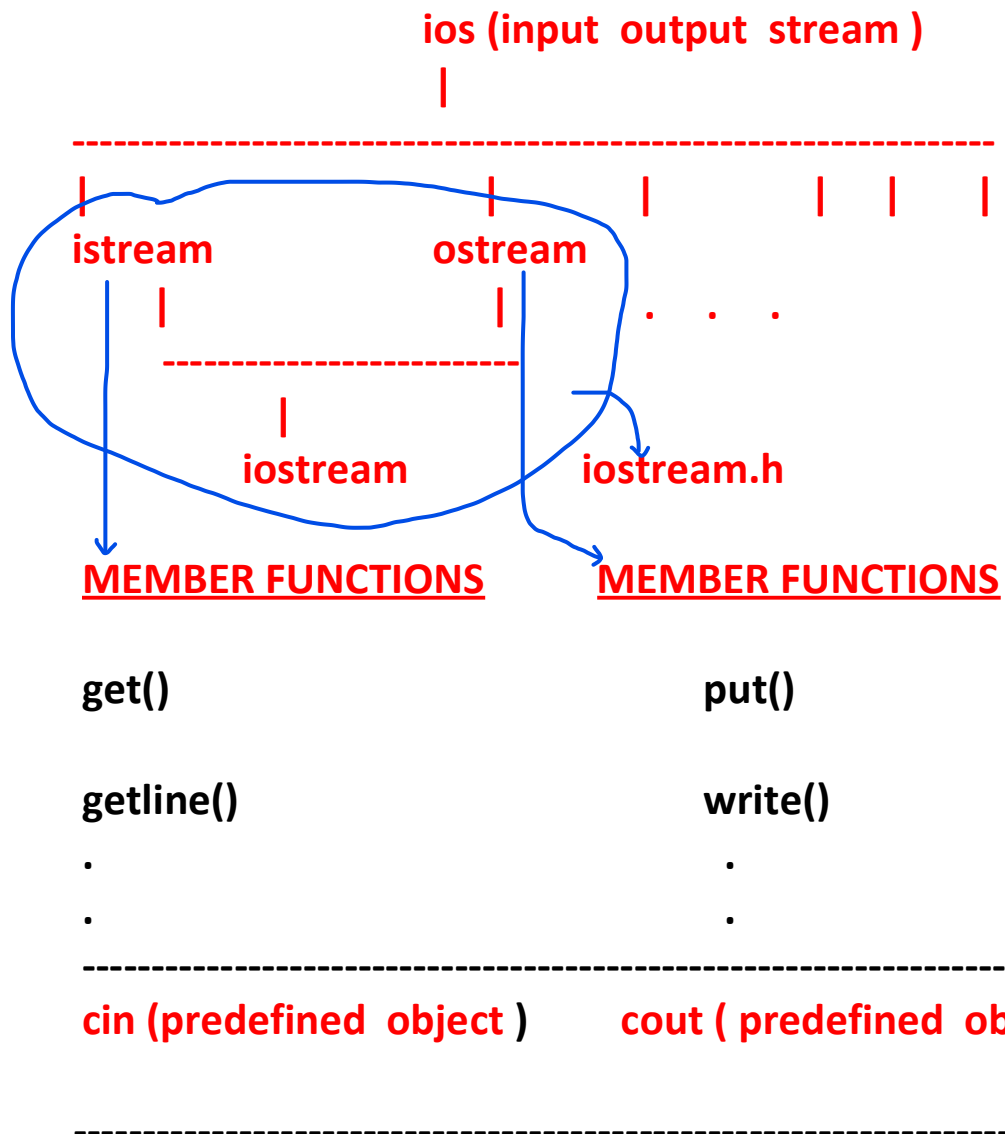


I/O (INPUT / OUTPUT)



A) UNFORMATTED I / O

cout<<a

1. << (INSERTION OPERATOR)

cin>>a

2. >> (EXTRACTION OPERATOR)

3. get() :- // INPUT SINGLE CHARACTER

syntax

variable = cin . get ();

4. put() :- // PRINT SINGLE CHARACTER

Syntax

cout . put(variable)

5. getline(): - // INPUT SPECIAL CHARACTER

syntax

cin . getline (variable , size);

6. write(): - // print special character

syntax

cout . write (variable, size);

Sameer Sir Classes, Jabalpur
Auth Exam Center Oracle, Microsoft
9407077858

// INPUT AND PRINT SINGLE CHARACTER

```
#include<iostream>
using namespace std;
```

```
int main()
{
    char ch;

    cout<< " ENTER CHARACTER = " << endl;
    ch = cin . get(); // A

    cout . put(ch); // A
}
```

// INPUT AND PRINT SPECIAL CHARACTERS

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
int main()
{
    char ch[80];

    cout<< " ENTER NAME = " << endl;
    cin . getline (ch,30); // raj kumar // gets(ch);
    cout. write (ch,30); // raj kumar // puts(ch);

}

cout . write ( ch , 5 ); // raj k
```

C LANG
↓

Sameer Sir Classes, Jabalpur
Auth Exam Center Oracle, Microsoft
9407077858

B) FORMATTED I/O

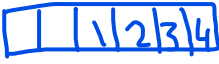
1. width() :- // RIGHT JUSTIFIED

syntax

cout . width (size);


E.G.

1. cout . width(6);
cout << 1234; // printf("%6d",1234);




A diagram showing a horizontal box divided into six equal-width segments. The first two segments are empty, and the last four segments contain the digits '1', '2', '3', and '4' respectively, illustrating right justification within a width of 6.

2. cout. width(6);
cout << 1234;
cout. width(5);
cout << 56; // printf("%6d%5d",1234,56);



A diagram showing two horizontal boxes. The first box is divided into six segments, with the last four containing '1', '2', '3', and '4'. The second box is divided into five segments, with the last two containing '5' and '6'. This illustrates the output of two formatted numbers with widths of 6 and 5.

3. cout.width(3);
cout << 12345; // printf("%3d",12345);



A diagram showing a horizontal box divided into five equal-width segments, each containing one of the digits '1', '2', '3', '4', and '5' from left to right, illustrating the output of a number with a width of 3.

Sameer Sir Classes, Jabalpur
Auth Exam Center Oracle, Microsoft
9407077858

2. precision() :-

syntax
cout . fill('*');
cout . precision(size);

↳ default +

3. fill():

syntax

cout.fill(character);

e.g. cout.width(8);

cout.precision(2);

cout.fill('*')



*	*	*	1	2	.	1	6
---	---	---	---	---	---	---	---

cout << 12.16412; // printf("%*8.2f", 1216412);

Sameer Sir Classes, Jabalpur
Auth Exam Center Oracle, Microsoft
9407077858

```
cout << 15.1356 ; // 15.14
```

```
cout << sqrt(3) ; // 1.73
```

```
cout<< 12.20004; // 12.2
```

```
imp
```

```
---
```

```
cout.width(3);
```

```
cout<<12345 ; // printf("%3d",12345);
```

```
ans :- 12345
```

Sameer Sir Classes, Jabalpur
Auth Exam Center Oracle, Microsoft
9407077858