```
printf ("Percentage = %.2f\n", Perc);
if (Perc >= 80)
{
printf ("Honours");
}
else if (Perc >= 70)
{
printf ("first class"); }
else if (Perc >= 60)
{ printf ("scond class"); }
else
{ printf ("fail") }.
```

Comparision Between C & C++ :-

C Header	C++
`#include <stdio.h>`	`#include <iostream>`
Input / output	
`scanf ("%d", &n);`	`cin >> age;`
`Printf ("Total", n);`	`cout << "Age is" << age.`

5-9-23

Structured Programming

Operators in C++

Arithmetic operator :-

+, -, *, %/0, /

Relational :-

>, <, ==, <=, >=, !=

logical :-

&&, ||, !

Assignment :-

=, +=, -=, .* =, /=/ %. =

Ternary :-

? :

C program :-

```
#include <stdio.h>
int main()
{
    int phy, chem, bio, math, sans;
    float perc.
    printf("enter 5 subjects marks;");
Scanf("%d %.d %.d %.d %.d", *phy, &chem,
    & bio, & math, & sans);
Perc = (Phy + chem + bio + math +
    sans) / 5.0;
```

```
printf ("Percentange : %.2f\n", perc);
if (perc >= 80)
{
printf ("Honours");
}
else if (perc >= 70)
{
printf ("First class"); }
else if (perc >= 60)
{ printf ("second class"); }
else
{ printf ("fail"); }.
```

Comparision Between C & C++ :-

C Header	C++
#include <stdio.h>	#include <iostream>
Input / Output's	
scanf ("%d", &n);	cin >> age;
printf ("Total %d", n);	cout << "Age is" << age.

if (condition)

{ printf ("Result");}

else

{ printf ("Recount");}

if ( )

{ printf "Result";}

else

{ printf "Recounting"}

Keywords :- (c)

auto	break	const
case	char	continue
default	do	double
else	enum	extern
float	for	goto
if	int	long
register	return	short
signed	sizeof	static
struct	switch	static
static	unsigned	void
volatile	while	

keywords (C++) :-

asm	double	new	switch
auto	else	operator	template
break	enum	private	this
case	explicit	protected	throw
catch	float	public	try
char	for	register	typedef
class	friend	return	
const	goto	short	
continue	if	signed	
default	inline	sizeof	
delete	int	static	
do	long	struct	
union			
unsigned			
virtual			
void			
volatile			

C statements

1) Input / Output

```
scanf ("%d", &n);
printf ("total", n);
```

2) If condition

```
If (condition)
{ printf ("Result"); }
else.
```

3) Loop(for);

```
for (int i=0; i<5; i++)
{
printf("Execution %d\n",i);
}
```

C++ statements :-

1) If condition

```
If (condition)
{cout<<"Result";}
else
{cout<<"Remaining";}
```

2) loop (for)

```
for (int i=1; i<=5; i++)
{
std::cout<<i<<" ";
}
```

3) variable declaration & initialization

```
int score =95;
int age =20;
```

Switch :- (c)

```
Switch (expression){
Case X;
   break;
default :
 int time =11;
 Switch (time){
 case 11
   cout "mrng";
  case 2;
   cout "Aft noon";
    break;
  }
```

for loop :-

```
for c printf; test; of date)
   { body}
   for c ; = 1; i<11; ++;
     { print ("%d",i)}
```

Programs:-

1) Adding two numbers:-

```
# include <iostream>
using namespace std;
int main(){
int n1, n2, sum;
cout << "Enter two num: ";
cin >> n1 >> n2;
sum = n1 + n2;
cout << n1 << "+" n2 << "=" << sum;
}.
```

2) swap two numbers-

```
# include <iostream>
using namespace std;
int main(){
int a=5, b=10, s;
cout << "Before swapping";
cout << "a=" << a << ", b=" << b;
    s=a;
    a=b;
    b=s;
cout << "After swapping";
cout << "a=" << a << ", b=" << b;
```

3) find larg number :-

```
#include <iostream>
using name space std;
int main() {
int a, b, c, ans;
ans = max(a, max(b, c));
cout << ans << " larg number";
}
```

4) odd or even :-

```
# include <iostream>
using namespace std;
int main(){
int a;
cout << " enter integer ";
cin >> a
if (a % 2 == 0){
cout << " given num is even"; }
else{
cout << " given num is odd"; }
}
```

5) Sum of natural numbers:-

```
#include <iostream>
using name space std;
int main(){
int n, s=0;
cout << "enter num;";
cin >>n;
for (int i=1; i<=n; ++i){
    s +=i;
}
cout<<"sum = "<<s;
}
```

afternoon time:- 12:15

1) Display fibonacci series:-

```
#include <iostream>
using namespace std;
int main() {
    int n;
    long factorial =1.0;
cout << "enter a positive integer
cin>>n;
```

```
if (n < 0)
    cout << "Error! factorial of a
            negative number doesn't
            exist.";
else {
    for (int i = 1; i <= n; ++i){
        factorial *= i;
    }
    cout << "factorial of " << n <<
        " = " << factorial;
}

return 0;
}
```

2) C++ program to Reverse a number:-

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

int fibonacci(int n){
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return (fibonacci(n-1) +
            (fibonacci(n-2));
```

```
}
int main() {
    int n;
    cout << "Enter the number
                of terms: ";
    cin >> n;
  cout << "fibonacci series: ";
  for(int i=0; i<n; i++){
    cout<< fibonacci (i) << ", ";
  }
  return 0;
  }
```

3) Check whether the number is prime or not :-

```
# include <iostream>
  using namespace std;
  int reverseDigits (int num) {
  static int reversed_number=0,
                remainder;
    if (num != 0) {
      remainder = num % 10;
    reversed_number = reversed_number
      * 10 + remainder;
      reverseDigits (num/10);
```

```
}
return reversed - number;
}
int main() {
    int n = 12345;
    cout << "Reversed number:
        " << reverseDigits(n);
    return 0;
}
```

4) Write a C++ program to multiply two numbers :-

```
#include <iostream>
#include <cmath>
using namespace std;
bool isPrime(int n) {
    if (n <= 1) {
        return false;
    }
    for (int i = 2; i <= sqrt(n);
                    ++i){
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}
int main() {
    int n;
```

```
cout << "Enter a positive integer: ";
cin >> n;
if (isPrime(n)) {
    cout << n << " is a prime number";
} else {
    cout << n << " is not a prime number";
}
return 0;
}
```

5)

```
#include <iostream>
using namespace std;
int main() {
    int num1, num2, product;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;
    product = num1 * num2;
    cout << "product = " << product;
    return 0;
}
```

6)

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

int main() {
    string str = "Hello World!";
    int length = str.length();
    cout << "The length of the string
        is: " << length;
    return 0;
}
```