DSA - Notes C++

Developers Club <VVCE>
Competitive Coding

1 Hello World

```
#include <iostream>
using namespace std;
int main() {
     cout<< "Hello world!" << endl;
     return 0;
}
```

Output:

```
Hello World!
```

2 Printing a shape

```
cout<< "   *" << endl;
cout<< "  **" << endl;
cout<< " ***" << endl;
```

Output:

```
    *
   * *
  * * *
```

3 Variables

```
String characterName = "John";
int characterAge = 35;

Cout<< "There was a man named "<< characterName;
cout<< " His age was" << characterAge;
```

Output

```
There was a man named John
His age was 35
```

4 Data types

```
char grade = 'A';
string phrase = "Hello";
int age = 50;
float gpa = 2.3;
bool isMale = false;
```

5 Working with Strings

```
string phrase = "Hello";
cout<< phrase [2];
```

Output

```
l
```

```
phrase [2] = 'x';
cout << phrase;
```

Output

```
Hexlo
```

```
cout<< phrase.find("lo", 0)
```

Output

```
3
```

```
cout<< phrase.substr (2, 4)
```

Output

```
xlo
```

$\boxed{6}$ Working with numbers

```
int n1 = 5;

cout << n1;
```

Output

```
5
```

```
float n2 = 5.5;

cout << n2;
```

Output

```
5.5
```

```
int a = 10, b = 20, c;
 c = a+b;
cout << c;
```

Output

```
30
```

```
c = a-b;
cout << c;
```

Output

```
-10
```

```
c = a*b;
cout << c
```

Output

```
200
```

```
c= a/b;
Cout<<c
```

output

```
0
```

if we take the c value in float we get 0.5

```
c = a%b;
Cout<<c
```

Output

```
1
```

```
a= 10
a+= 10;
cout<<a
```

Output

```
20
```

```
a-=10
cout<<a
```

output

```
0
```

Same can be done for * and /

```
#include <cmath>

int a = 3;
cout << pow (a,3);
```

Output

```
27
```

```
int a = 9;
cout << sqrt(a);
```

output

```
9
```

```
cout << ceil (4.1);
```

output

```
5
```

```
cout << floor (4.1);
```

output

```
4
```

```
cout << fmin (9,10);
```

output

```
9
```

```
cout << fmax (9,10);
```

output

```
10
```

Scanned with CamScanner

7 Getting User input

```
cout<< " Enter your age: ";
cin>> age;
cout<<" You are "<<age<< " years old";
```

Output

```
Enter your age: 30
You are 30 years old
```

8 Building a calculator

```
int num1, num2;
cout<< "Enter first number";
cin>> num1;
cout<<" Enter second number";
cin>>num2;
cout<< num1 << " + " << num2 << " = " << num1+num2;
```

Output

```
Enter first number : 20
Enter second number: 30
20 + 30 = 50
```

9 Arrays

① Creating an array

```
int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

② Accessing an element

```
int elem = arr[5];
cout << elem;
```

output

```
6.
```

③ Using a loop

```
for (int i=0; i<10; i++)
    cin>>a[i];

for (int i=0; i<10; i++)
    cout << a[i];
```

10 Functions

~~int add~~

```
void greet() {
    cout<< "Hello!";
}
int main() {
    greet();
}
```

output

```
Hello!
```

Other types of functions:

```
int sum (int a, int b) {
    int c= a+b;
    return c;
}

int main () {
    int a = 10; b=20;
    sum (a,b);
}
```

output

```
30
```

II Conditional statements

① If statement

```
if (age >18) {
    cout << "Eligible to vote"
}
```

② else and else if

```
if (grade == 'A') {
    cout<< "Outstanding"
} else if (Grade == 'B')
    cout << "Excellent";
else
    cout << "Poor Performance";
```

12 Operators.

==	equal
!=	not equal
>	Greater than
<	less than
>=	Greater than or equal to
<=	less than or equal to
&&	AND
||	OR
!	NOT

```
if(a==b)  (a!=b)  (a>b)  (a<b)  (a>=b) (a<=b)

int a = 10;
int b = 0;

cout << (a==b)
          :
          :
        (a<=b)
```

```
bool x = true;
bool y = false;

if (x && y) {
    cout << "Hey!";
} else {
    cout << "Hello";
}
```

Test for || and ! also.

13 Building a better Calculator

+ - * /

```
int num1, num2;

Char op;

Cout << " Enter num1";

Cin >> num1;

Cout << " Enter num2";

Cin >> num2;

Cout << " Enter +, -, *, / ";

Cin >> op;

if ( op == '+')
  cout << num1 + num2;
if (op == '-')
  cout << num1 - num2;
if (op == '*')
  cout << num1 * num2;
if (op == '/')
  cout << num1 / num2;
```

1h Switch statements

```
String getDayOfWeek ( int dayNum) {
        String dayName;
        if( dayNum == 0) {
            dayName = "Sunday";
        } else if( dayNum == 1) {
            dayName = "Monday";
        }
              :
              :
```

using switch statement

```
String getDayOfWeek ( int dayNum) {
        switch (dayNum) {
            case 0 :  cout << "Sunday";
                      break;
            case 1 :  cout << "Monday";
                      break;
                   :
            case 6:  cout << "Saturday";
                     break;
            default: cout << "Please enter a valid number";
}
```

15 Loops

① While loop

```
int i=0;
while (i<10) {

    cout<<i<<" ";
    i++;
}
```

② do-while loop

```
do {
   cout<<i<<" ";
   i++;
}while (i<10);
```

③ for loop

```
for (int i=0; i<10; i++)
    cout<<i<<" ";
```

16 2D array (Matrix)

```
int a[3][3];

for (i=0; i<3; i++)
    for(j=0; j<3; j++)
        cin>>a[i][j];
```

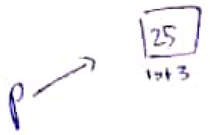
	0	1	2
0			
1			
2			

17 Comments

```
// This is a comment
/* This
   is
   a
   multiline
   comment
*/
```

18 New and Delete

```
int *p = new int(25);
cout << *p;
delete p;
```



19 Pointers

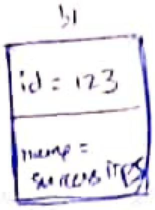
```
int age = 19
cout << &age;

int a = 10;
int *b;
b = &a;
cout << b;
```

20 Classes and Objects

Structure

```
struct book {
  int bookid;
  string bookname;
};
```

```
int main ()
{  book b1;
   b1. id = 123;
   b1. bame = " Success Tips";
```



-> Difference between class and structure

Structure cannot have functions

Class can have functions also.

```
class Book () {

    ~~book~~ int id;
        string name;
    void set_id (Book, id) {
        Book. id = id
    }
    ~~void~~ int get_id ( ~~Book id~~ Book) {
        return Book. name;
    }
};

int main () {

   Book b1;
   b1: set_id (b1, 1);
   Cout << b1. get_id;
}
```

21 Constructor and Destructor

```
class Book {
    public:
        string title;
        string author;
        int pages;
};

int main()
{
    Book b1;
    b1. title = " Good Days";
    b1. author = " JK Rowling";
    b1. pages = 500;
```

b1

title	Good days
author	JK Rowling
pages	500

```
    Book b2;
    b2. title = " Geometry";
    b2. author = " Paulo";
    b2. pages = 200;
}
```

b2

title	Geometry
author	Paolo
pages	200

Using constructor this process can be simplified:

What is constructor?

```
→ class Book {
      public:
          string title;
          string author;
          int pages;
          Book() {
            cout << " Book added";
          }
  };
```

We can use parameters also:

```
class Book {
        :
    Book(string Title, string Author, int Pages) {
        title = Title;
        author = Author;
        pages = Pages;
    }
};

int main() {
    Book b1 ("Good days", "JK Rowling", 100);
}
```

[iv] Inheritance

```
class Vehicle {
    int id;
    string name;
};
class car : public Vehicle {
    string model;
};
int main()
{
    Vehicle V1;
        V1.id = 1;
        V1.name = "Toyota";
        Car C1;
        C1.model = "hybrid";
}       C1.name = "Nissan";
```