NOTES ON CLASS 1 – BASICS OF C++

Note:- Some practice problems might have a option to solve only in java such problems can be solved by CPP at your end.

1. C++ User Input

In CPP,

- cout is used to output (print) values.
- cin to get user input.

cin is a predefined variable that reads data from the keyboard with the extraction operator (>>).

Example

```
int x;
cout << "Type a number: "; // Type a number and press enter
cin >> x; // Get user input from the keyboard
cout << "Your number is: " << x; // Display the input value
```

cout is a predefined variable which uses the << operator, is used to output values/print text:

Example

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

int main() {
  cout << "Hello World!";
  return 0;
}
```

2. Datatypes

Name	Description	Size*	Range*
char	Character or small integer.	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)

3. If-else

```
if(condition1==true)
{
    //this block will be run if condition1 is true
}
else if(condition2==true)
{
    //this block will run if condition2 is true
}
else
{
    //this block will run otherwise
}
```

Practice Problem :-

https://practice.geeksforgeeks.org/problems/java-if-else-decision-making0924/0?category%5B%5D=Java&category%5B%5D=Java&difficulty%5B%5D=-2&page=1&query=category%5B%5DJavadifficulty%5B%5D-2page1category%5B%5DJava

4. Typecasting

Typecasting: It is a datatype is converted into another data type by the programmer using the casting operator during the program design.

```
#include <bits/stdc++.h>
// Function to divide a and b
void division(int a, int b)
{
   float div;
   // Typecasting in float
   div = (float)a / b;
   cout<<"The result is “<<div<<”\n";
}
int main()
{
   // Given a & b
   int a = 15, b = 2;
   // Function Call
   division(a, b);
   return 0;
}
```

Output:

The result is 7.500000

5. Switch Statement

The syntax for a switch statement in C++ is as follows −

```
switch(expression) {
  case constant-expression  :
    statement(s);
    break; //optional
  case constant-expression  :
    statement(s);
    break; //optional
  // you can have any number of case statements.
  default : //Optional
    statement(s);
}
```

Practice Problem :

https://practice.geeksforgeeks.org/problems/java-switch-case-statement3529/1

6. For Loop

When you know exactly how many times you want to loop through a block of code, use the for loop instead of a while loop:

Syntax

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

The example below will print the numbers 0 to 4:

Example

```
for (int i = 0; i < 5; i++) {
  cout << i << "\n";
}
```

7. While Loop

The while loop loops through a block of code as long as a specified condition is true:

Syntax

```
while (condition) {
  // code block to be executed
}
```

In the example below, the code in the loop will run, over and over again, as long as a variable (i) is less than 5:

Example

```
int i = 0;
while (i < 5) {
  cout << i << "\n";
  i++;
}
```

Practice Problem :

https://practice.geeksforgeeks.org/problems/while-loop-printtable-java/1

8. Function(Pass By Reference and Pass By Value)

Pass by Value

In pass by value, the actual value that is passed as argument is not changed after performing some operation on it. When call by value is used, it creates a copy of that variable into the stack section in memory. When the value is changed, it changes the value of that copy, the actual value remains the same.

Example Code

```
#include<bits/stdc++.h>
```

```
using namespace std;

void my_function(int x) {
  x = 50;
  cout << "Value of x from my_function: " << x << endl;
}

int main() {
  int x = 10;
  my_function(x);
  cout << "Value of x from main function: " << x;
return 0;
}
```

Output

Value of x from my_function: 50

Value of x from main function: 10

Pass by Reference

In pass by reference the actual value that is passed as argument is changed after performing some operation on it. When call by reference is used, it creates a copy of the reference of that variable into the stack section in memory. Is uses a reference to get the value. So when the value is changed using the reference it changes the value of the actual variable.

Example Code

```
#include<bits/stdc++.h>
using namespace std;

void my_function(int &x) {
  x = 50;
  cout << "Value of x from my_function: " << x << endl;
}
```

```
int main() {
  int x = 10;
  my_function(x);
  cout << "Value of x from main function: " << x;
  return 0;
}
```

Output

Value of x from my_function: 50

Value of x from main function: 50

Where to use Pass by reference?

- The pass by reference is mainly used when we want to change the value of the passed argument into the invoker function.
- One function can return only one value. When we need more than one value from a function, we can pass them as an output argument in this manner.

😊 KEEP CODING AND KEEP GROWING 😊