Day: Date: / /

Data types:

1) Integer data types:

- The primary consideration while choosing a numeric data type are:
  - The largest and smallest numbers that can be stored in a variable
    ~~(If we exceed)~~ (If we exceed the limit, then the smallest or largest number replaces the number).
  - How much memory the variable uses.
  - Variable stores signed or unsigned.
  - number of decimal place precision.

Datatype	Size	Range
Short Int	2B	$-2^{15} - (2^{15}-1)$
Unsigned short int	2 B	$0 - (2^{16}-1)$
int	4B	$-2^{31} - (2^{31}-1)$
unsigned int	4B	$0 - (2^{32}-1)$
long int	4B	$-2^{31} - (2^{31}-1)$
unsigned long int	4B	$0 - (2^{32}-1)$
long long int	8 B	$-2^{63} - (2^{63}-1)$
unsigned long long int	8B	$0 - (2^{64}-1)$

If we only write long or long long without int then it ~~can~~ will still be ~~used~~. considered int.

1 subtracted as 0 is also considered: in the ri

What if we go above the limit of ~~an~~ variable?

```
#Include <iostream>
Using namespace std;

Int main()
{
    ~~int~~ int a= 2000 000 000;
        cout << "The value of a: "<< a <<endl;

    return 0;
}
```

Terminal

```
The value of a: 2147483647
```

The value of a decreased!!!

- The max value int can store is $(2^{31}-1)$ which
- is 2,147,483,647. So the value changes to the max value stored.

How to ~~[illegible]~~ find size of variable:

Sizeof()

We can we the built-in function of C++: sizeof( )

e.g

cout << sizeof (int) << sizeof (a);

both will give answer of 4 Bytes.

Char data types:

This is used to store individual characters.
It can only hold one character at a time.

example: (consider the int main and return are already called).

```
char letter;
letter = 'g';
```

} This is how to ~~to~~ write a single char.
We will not use "" as we are not assigning a string and it will give an error.

- Char data type is typically only one Byte or 8 bits i.e $2^8 = 256$.

char	1B	-128 – 127
unsigned char	1B	0 – 255

Char is an int data type !?!

- Although char is used to store characters, it is actually an int data type :
- The reason that it is used to store characters is that character internally represent numbers.
- charatters are encoded using mostly ASCII.
- There are 256 numbers in one byte and every number store a different character.
- So there are 256 different chars.

example: (intmain and return already written).

```
char letter=65;
cout << letter << endl;
 letter = 66
cout << letter << endl;
letter = 97
cout << letter << endl;
```

from 65, Capital letters start. from 97, samaller letter start.

Terminal

```
A
B
a
```

65 represent A
66 is B
97 is a.

3) Floating point Datatypes:

- Used to store decimal numbers.

float	4 B	6 Digits
Double	8 B	15 Digits
long Double	8 B	18 Digits

Precising the amount of digits after d.p:

- Normally c++ compilers represent the number by 6 digit only e.g (33.3333). but we can change this.

- We can use set precision () from <iomanip> to change precision:

Int main and return already written:

```
float x; double y;
x = 10.0/3 ; y = 10.0/3 ;

cout<< setprecision (10) << x<<endl<<y <<endl;
```

garbage values

```
3.333333524 → x
3.333333333 → y
```

x was a float and it can only have 7 dig. y was double and it can have 15 dig.

Day: Date: / /

The previous programme showed that the ~~[illegible]~~ all digits add upto 10 but if we want e.g 10 digit after decimal then:

3) `cout << fixed << setprecision (10) << y << endl;`

This will allow 10 digits after d.p.

What if we did 10/3 instead of 10.0/3 ?

~~[illegible]~~ . Mathematical operation happen first.

10 → int
3 → ~~[illegible]~~ int = 3 , so answer also int.

answer would have been 3.0 instead of 3.333 ___

so we used

10.0 → float = 3.333
3 → ~~float~~ int

even if one is float, the answer is float.

4) bool ~~oper~~ data type:

bool can have two values only, 1 or 0, True or False.

bool	1 bit	0/1, True/False

5) String Data type:

For this ve will require a librabry:

<string>

Detail will be discussed later.