Tutorial 2: Basics of C

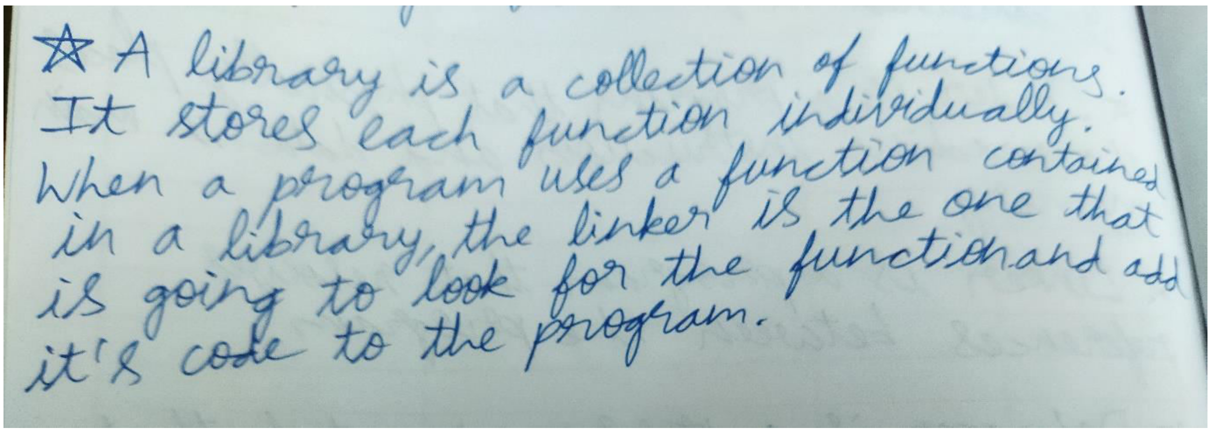
1. Differentiate between compiling and linking?

- A compiler takes the file containing source code and translates it to machine code. It then places that machine code into an output file called an object file and has the file extension .obj. The process of translating the source code into an object file is called compiling.
- After the compiler has created all the object files, another program is called to bundle them into an executable program file. That program is called a linker and the process of bundling them into the executable is called linking.

2. Explain the purpose of header file in C?

The purpose of a header file is to propagate declarations to code files. A header file is a file with extension . h which contains C function declarations and macro definitions to be shared between several source files.

3. Explain the purpose of using library in C?



☆ A library is a collection of functions. It stores each function individually. When a program uses a function contained in a library, the linker is the one that is going to look for the function and add it's code to the program.

Library functions in C language are inbuilt functions which are grouped together and placed in a common place called library. Each library function in C performs specific operation. We can make use of these library functions to get the pre-defined output instead of writing our own code to get those outputs.

Samuela Abigail
71762108039

4. Mention one difference between declaring a variable and defining a variable?

Definition of a variable says where the variable gets stored

Declaration gives details about the properties of a variable.

5. State the reason for data type specified for variable declaration? And to identify the amount of memory needed by the variable

Data type is the kind of information the variable can hold. This determines how the program/computer can use that information.
Identify the type of a variable when it declared. Identify the type of the return value of a function. Identify the type of a parameter expected by a function.

6. Explain the uses of void in C?

- When void is used as a function return type, it indicates that the function does not return a value.
- The void data type is typically used in the definition and prototyping of functions to indicate that either nothing is being passed in and/or nothing is being returned.
- When void appears in a pointer declaration, it specifies that the pointer is universal. void pointers should be used any time the contents of a block of data is not important.
- For example when copying data the contents of a memory area is copied but the format of the data is not important

7. Which declaration is correct : main(0 or void main () or int main (), explain based on your understanding whether main() must?

- main (0 is an error.
- void main () and int main () are correct.
- Yes, main () is a must. The main function is a primary function which serves as the starting point for program execution. It usually controls program execution by directing the calls to other functions in the program.

8. Should main () always a return a value?

In a main function, the return statement and expression are optional.The return value of main() function shows how the program exited. The normal exit of program is represented by zero return value. If the code has errors, fault etc., it will be terminated by non-zero value.

9. How can you check what value is returned from main() ? is the executed program terminated normally or not?

- It can check through batch file or shell script.
- The main() program returns a zero if executed successfully and one if exited with error.
- The main() program can be coded in such a way to return different values for different types of error.

10.Classify the different type of operators in C ?

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Conditional Operators
- Special Operators

11. Explain the difference of lvalue and rvalue?

lvalue-

- An lvalue is an expression to which a value can be assigned .
- The lvalue expression is located on the left side of an assignment statement.
- The address associated with a program variable in C is called its lvalue

Rvalue-

- An rvalue can be defined as an expression that can be assigned to an lvalue.
- An rvalue is located on the right side of an assignment statement.
- The contents of lvalue location are its rvalue, the quantity that is supposed to be the value of the variable.

12. Why does the statement a+b = c+d is not legal in C?

The left side of the statement evaluates to a constant value which cannot be changed and it does not represent a storable location in the memory. Therefore the assignment statement will not generate any value and will show compiler error.

13. Why should use i++ instead of i=i+1?

These two are exactly the same. It's just two different ways of writing the same thing. i++ is just a shortcut for i += 1 , which itself is a shortcut for i = i + 1 . These all do the same thing, and it's just a question of how explicit we want to be.

14. Can apply ++ and -- operators on floating point numbers? Mention the difference between prefix and postfix forms of the ++ operator?

- The ++ and -- operators are defined for all integral and floating-point numeric types, and hence can be applied on floating point numbers.
- The prefix operator ++ adds one to its operand / variable and returns the value before it is assigned to the variable.
- The postfix operator ++ adds one to its operand / variable and returns the value only after it is assigned to the variable.

15. Mention the precedence of operator?

1. `++` and `--` (increment and decrement) operators hold the **highest precedence**.
2. Then comes `-` (unary minus) operator
3. Then comes `*`, `/` and `%` holding equal precedence.
4. And at last, we have the `+` and `-` operators used for **addition** and **subtraction**, with the **lowest precedence**.

16. Explain the term cast refer to ? why it is used?

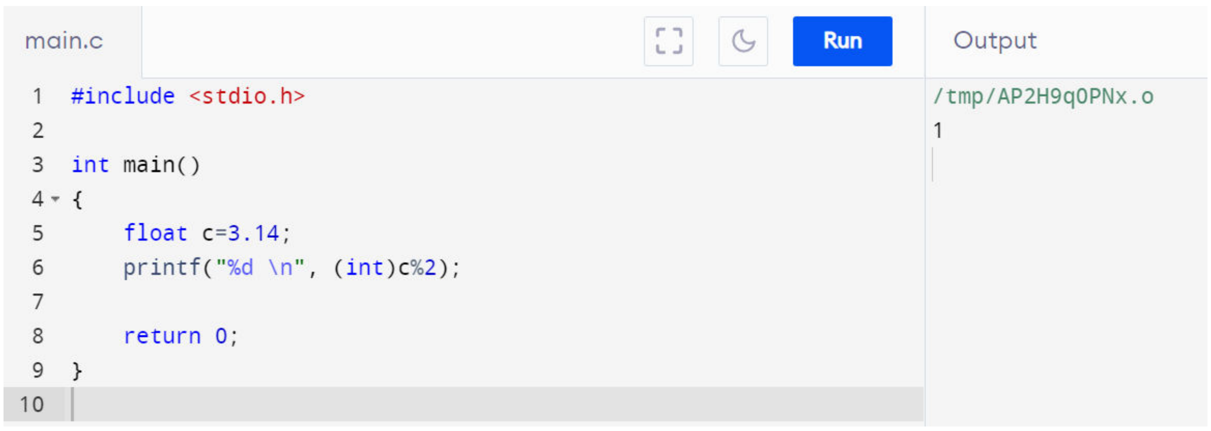
- Type casting refers to changing an variable of one data type into another.
- The compiler will automatically change one type of data into another if it makes sense. For instance, if you assign an integer value to a floating-point variable, the compiler will convert the int to a float. Casting allows you to make this type conversion explicit, or to force it when it wouldn't normally happen.

What is the output for following codes?

```
a)int main()
{
float c=3.14;
printf("%f", c%2);
return 0;
}
```

error: invalid operands to binary %

Corrected output:



```c
#include <stdio.h>

int main()
{
    float c=3.14;
    printf("%d \n", (int)c%2);

    return 0;
}
```

```
/tmp/AP2H9q0PNx.o
1
```

b)Int main ()

{

C=--2;

printf(”c=%d”, c)

return 0;

}

Output is error since C language is case sensitive and int is written as Int .Also, c is undeclared and lvalue required as decrement operand.

c)int main()

{

int x= 1, y=5;

prinff (“%d”, ++x+y);

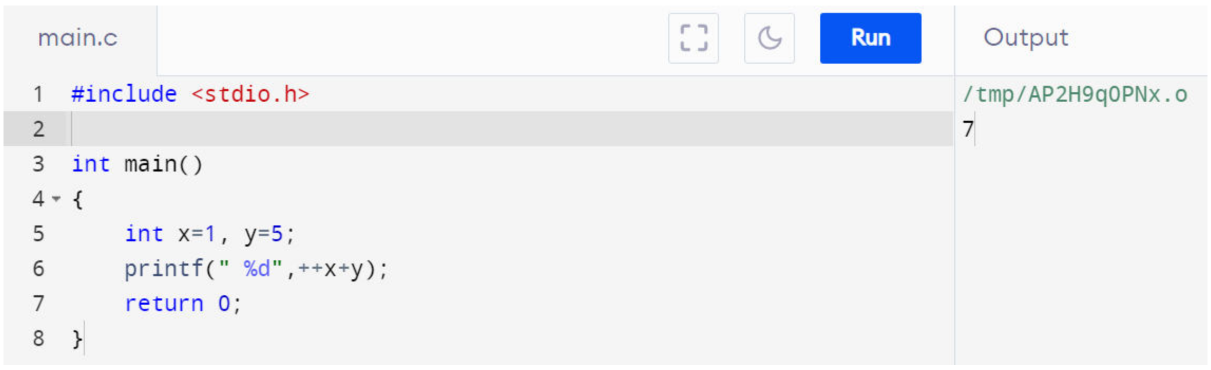
return 0;

}

Output is error since printf is misspelled as prinff

Samuela Abigail
71762108039

Corrected output:



```c
#include <stdio.h>

int main()
{
    int x=1, y=5;
    printf(" %d",++x+y);
    return 0;
}
```

```
/tmp/AP2H9q0PNx.o
7
```

Samuela Abigail
71762108039