**Name – Senuri Perera**

Student no. – 29779

Batch no. – 23.1

Degree – Software Engineering

**Tutorials**

Tutorial 1

1.Briefly explain the need of a programming language?

* **Programming languages allow humans to give instructions to computers in a readable and understandable manner.**

2. Compare and contrast the differences between followings;

1. Source Code vs. Machine Code

* **Source Code is a human-readable form of a program written in a specific programming language. Machine code also known as object code is a low level binary representation of instructions that a computer’s processor can directly execute.**
* **Readability- source code is readable and easily understandable by programmers but machine code is not designed to be human readable or easily understood.**
* **Execution- Source code need to be compiled or interpreted before compilation but machine code executed directly by computer’s hardware without translation or interpretation.**

1. High Level Language vs. Low Level Language

* **High level languages provide higher level of abstraction, readability and productivity while low level languages offer more control and efficiency at the cost of reduced readability and portability. High level languages more suitable for most software development tasks while low level languages are often used for system level programming,device drivers or performance critical applications.**

1. Compiler vs. Interpreter

* **Compiler translates the entire code at once while the interpreter but the interpreter translates the code line by line.**

1. Structured Language vs. Object Oriented Language

* **Structured Language is used to develop programs using modules or functions while object oriented language constructs the program using a set of objects.**

1. C vs. C++

* **C and C++ are high level languages. C is a structured language and C++ is a object oriented language. Main difference is C doesn’t support classes and objects.**

1. C++ v. Java

* **C and Java are high level languages and the main difference is C++ uses only compiler while Java uses both compiler and interpreter.**

1. Syntax error vs. Logical error

* **A syntax error occurs when the code violates syntax rules of the programming language while a logical error occurs when the code doesn’t produce the expected result due to flaws in the program’s logic. Syntax errors detected by compilation or interpretation and can be resolved by correcting the code’s syntax while logical errors require analysis and debugging to identify and fix the flawed logic in the code.**

Tutorial 2

1. How do you write comments in a c program? What is the purpose of comments in a program?

* **There are 2 types of ways to write comments in C program**

1. **Single line comment - //**
2. **Multi line comment - /\*……\*/**

* **Comments are used to explain the code better and make it more readable and understandable for humans.**

1. Which is the function that is essential in a C program?

* **main()**

1. What is the purpose of ‘scanf’ ?

* **To read formatted input from the user or a data stream.**

1. Is ‘standard c’ a case sensitive language?

* **Yes**

5. Determine which of the following are valid identifiers. If invalid, explain why.

(a) record1 **Valid**

(b) 1record **Invalid(Identifiers cannot start with a digit)**

(c) file-3 **Invalid(Identifiers cannot contain special characters like hyphens)**

(d) return **Valid**

(e) $tax **Valid**

(f) name **Valid**

(g) name and address **Invalid (Identifiers cannot contain spaces)**

(h) name-and-address **Invalid (Identifiers cannot contain hyphens)**

(i) name\_and\_address **Valid**

(j) 123 - 45 - 6789 **Invalid(Identifiers cannot have spaces and hyphens)**

6. State whether each of the following is true or false. If false, explain why.

a) Function printf always begins printing at the beginning of a new line.

* **False, because we use \n to start or break into a new line.**

b) Comments cause the computer to print the text enclosed between /\* and \*/ on the screen when the program is executed.

* **False, because comments are being ignored by the compiler and they do not produce output when the program is executed.**

c) The escape sequence \n when used in a printf format control string causes the cursor to position to the beginning of the next line on the screen.

* **True**

d) All variables must be defined before they’re used.

* **True**

e) All variables must be given a type when they’re defined.

* **True**

f) C considers the variables, number and NuMbEr to be identical.

* **True**

g) A program that prints three lines of output must contain three printf statements.

* **False, because we can print all three lines using one printf statement as follows: printf(“output\n output\n output\n”)**

7. What does the following code print?

printf( "\*\n\*\*\n\*\*\*\n\*\*\*\*\n\*\*\*\*\*\n" );

**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***

8. Identify and correct the errors in each of the following statements. (Note: There may be more than one error per statement.)

a) scanf( "d", value );

**scanf(“%d”,&value);**

b) printf( "The product of %d and %d is %d"\n, x, y );

**printf(“The product of %d and%d is %d \n”,x,y);**

c) Scanf( "%d", anInteger );

**scanf(“%d”,&anInteger);**

d) printf( "Remainder of %d divided by %d is\n", x, y, x % y );

**printf( "Remainder of %d divided by %d is %d", x, y, x % y );**

e) print( "The sum is %d\n," x + y );

**printf(“The sum is %d\n”,x+y);**

f) Printf( "The value you entered is: %d\n, &value );

**printf(“The value you entered is: %d\n”,value);**

9. What, if anything, prints when each of the following statements is performed? If nothing prints, then answer “Nothing.” Assume x = 2 and y = 3 .

a) printf( "%d", x );

**2**

b) printf( "%d", x + x );

**4**

c) printf( "x=" );

**x=**

d) printf( "x=%d", x );

**x=2**

e) printf( "%d = %d", x + y, y + x );

**5=5**

f) z = x + y;

**nothing**

g) scanf( "%d%d", &x, &y );

**nothing**

h) /\* printf( "x + y = %d", x + y ); \*/

**nothing**

I ) printf( "\n" );

**nothing**

10. State which of the following are true and which are false. If false, explain your answer.

a) C operators are evaluated from left to right.

* **False, operators evaluate from right to left**

b) The following are all valid variable names: \_under\_bar\_ , m928134 , t5 , j7 , her\_sales , his\_account\_total , a , b , c , z , z2 .

* **True**

c) The statement printf("a = 5;"); is a typical example of an assignment statement.

* **False, cannot assign inside inverted comas**

d) A valid arithmetic expression containing no parentheses is evaluated from left to right.

* **False,it will be according to the precedence.**

e) The following are all invalid variable names: 3g , 87 , 67h2 , h22 , 2h

* **False, variables cannot start with numbers except h22 all are invalid variables.**

Tutorial 3

Q1. Write four different C statements that each add 1 to integer variable x.

* **X+=1**
* **X++**
* **X=X+1**
* **++x**

Q2. Write a single C statement to accomplish each of the following:

1. Assign the sum of x and y to z and increment the value of x by 1 after the calculation.

* **Z=(x+y),x++;**

1. Multiply the variable product by 2 using the \*= operator.

* **\*=2**

1. Multiply the variable product by 2 using the = and \* operators.

* **product=product\*2**

1. Test if the value of the variable count is greater than 10. If it is, print “Count is greater than 10.”

* **If(count>10)**

**printf(“Count is greater than 10 \n”);**

1. Decrement the variable x by 1, then subtract it from the variable total.

* **Total -= --X;**

1. Add the variable x to the variable total, then decrement x by 1.

* **Total += X--;**

1. Calculate the remainder after q is divided by divisor and assign the result to q. Write this statement two different ways.

* **q%=divisor**
* **q = q%divisor**

1. Print the value 123.4567 with 2 digits of precision. What value is printed?

* **printf(“%.2f”,num);**

1. Print the floating-point value 3.14159 with three digits to the right of the decimal point. What value is printed?

* **printf(“%.3f”,num);**

Q3. Write single C statements that

1. Input integer variable x with scanf.

* **scanf(“%d”,&x);**

1. Input integer variable y with scanf.

* **scanf(“%d”,&y);**

1. Initialize integer variable i to 1.

* **int i=1;**

1. Initialize integer variable power to 1.

* **power = 1;**

1. Multiply variable power by x and assign the result to power.

* **power = power\*x;**

1. Increment variable i by 1.

* **i++;**

1. Test i to see if it’s less than or equal to y in the condition of a while statement.

* **while(i<=y)**

1. Output integer variable power with printf.

* **printf(“%d”,power);**

Tutorial 4

1. What is wrong with the following if statement (there are at least 3 errors). The Indentation indicates the desired behavior.

if numNeighbors >= 3 || numNeighbors = 4

++numNeighbors;

printf("You are dead! \n " );

else

--numNeighbors;

**If (numNeighbours>=3 || numNeighbours==4)**

**{**

**++numNeighbours;**

**printf(“You are dead!\n”);**

**else**

**{**

**--numNeighbours;**

**}**

1. Describe the output produced by this poorly indented program segment:

int number = 4;

double alpha = -1.0;

if (number > 0)

if (alpha > 0)

printf("Here I am! \n" );

else

printf("No, I’m here! \n");

printf(“No, actually, I’m here! \n");

**Output:**

**No, I’m here!**

**No, actually, I’m here**

1. Consider the following if statement, where doesSignificantWork, makesBreakthrough,

and nobelPrizeCandidate are all boolean variables:

if (doesSignificantWork) {

if (makesBreakthrough)

nobelPrizeCandidate = true;

else

nobelPrizeCandidate = false;

}

else if (!doesSignificantWork)

nobelPrizeCandidate = false;

**No output as question is not clear.**

1. Write if statements to do the following:

– If character variable taxCode is ’T’, increase price by adding the taxRate percentage of price to it.

**If(taxCode==’T’) {**

**Total price = price+(taxrate/100)\*price**

**}**

– If integer variable opCode has the value 1, read in double values for X and Y and calculate and print their sum.

**If(opcode==1) {**

**sum=(double)x+y;**

**Printf(“%.2f)”,sum);**

**}**

– If integer variable currentNumber is odd, change its value so that it is now 3 times currentNumber plus 1, otherwise change its value so that it is now half of currentNumber (rounded down when currentNumber is odd).

**If(currentNumber%2==1) {**

**currentNumber = (currentNumber\*3)+1;**

**else{**

**currentNumber = currentNumber/2;**

**}**

– Assign true to the boolean variable leapYear if the integer variable year is a leap year. (A leap year is a multiple of 4, and if it is a multiple of 100, it must also be a multiple of 400.)

**if(year%4==0) || (year%100==0) && (year%400==0) {**

**leapyear=true;**

**}**

– Assign a value to double variable cost depending on the value of integer variable distance as follows:

**Distance Cost**

----------------------------------- ----------

0 through 100 5.00

More than 100 but not more than 500 8.00

More than 500 but less than 1,000 10.00

1,000 or more 12.00

**If(distance>=0 && distance <=100) {**

**Cost = 5;**

**}**

**Else if(distance>100 && distance<500) {**

**Cost = 8;**

**}**

**Else if(distance>500 && distance<1000) {**

**Cost = 10;**

**}**

**Else {**

**Cost=12;**

**}**

Tutorial 5

**Switch**

Input two numbers and display the outputs of the basic mathematic operations. The output screen should be displayed as follows;

Enter two numbers \_\_\_\_ \_\_\_\_

|  |
| --- |
| 1. + |
| 2. – |
| 3. \* |
| 4. / |
|  |
| Please enter your Choice \_\_\_ |
|  |

A screenshot of a computer program

Description automatically generated with medium confidence

**While loop**

1. Input 10 numbers and display the total count of odd & even numbers in the entered number series.

2. Modify the above program in to enter series of numbers terminates when the user enter -99 and display the same expected output

1.

A screenshot of a computer code

Description automatically generated with medium confidence

2.

A screenshot of a computer program

Description automatically generated with medium confidence

**Do while loop**

Rewrite the programs for the above while loop question 1 & 2 using do while loop

A screenshot of a computer code

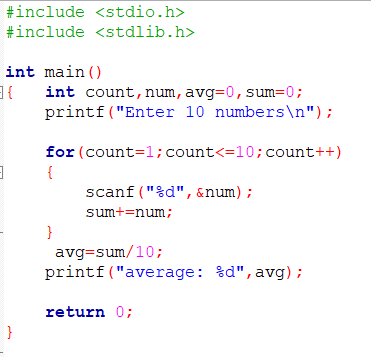
Description automatically generated with medium confidence1.

A screenshot of a computer program

Description automatically generated with medium confidence2.

**For loop**

1. Input 10 numbers and display the average value using the for loop



2. Display the following output using the for loop

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

A screenshot of a computer code

Description automatically generated with medium confidence