**Tutorial 1 to 5**

**Name: Hiruni Ganege**

**ID Number: 30197**

**Tutorial 01**

**Question 01 (answer):**

Programming languages use classes and functions that control commands. The reason that programming is so important is that it directs a computer to complete these commands over and over again, so people do not have to do the task repeatedly. Instead, the software can do it automatically and accurately.

**Question 02 (answer):**

1. **Source cord vs machine code**

Source code

• Programming code that has not yet been compiled into executable file .

• Source cord is written in high-level or assembly language.

Machine code

• Compiled code that can be run is any computer with same CPU architecture.

• machine code is any low-level programming language.

• Each instruction causes the CPU to perform a very specific task, such as a load, a store, a jump, or an arithmetic logic unit (ALU) operation on one or more units of data in the CPU's registers or memory.

**(B) High-Level Language vs. Low-Level Language**

High-Level Language

• High-Level languages are easy to lean and understand.

• They allow much more abstraction

• They do not provide many facilities at hardware level

Low-Level Language

• Low level languages are challenging to learn and understand.

• They allow little or no abstraction

**(C) Compiler vs. Interpreter**

Compiler

• Fast , creates executable file that runs directly on the CPU

• Debugging is more difficult. One error can produce many false errors.

• More likely to crash the computer. The machine code is running directly on the CPU

EX - C , C++ , C# , java

Interpreter

• Slower , interprets code one line at a time

• Debugging is easier . Each line of code is analysed and checked before being executed.

• Uses less memory, source code only has to be present one line at a time in memory.

EX - python , PHP , Ruby , Perl

**(D) Structured Language vs. Object-Oriented Language**

Structured Language

• It deals with algorithm.

• Program are divided into function .

• The program is represented as a logical structure.

• Lees date security.

Object-Oriented Language

• It deals with date.

• Program are divided into object.

• The program is written as a collection of objects which communicate with each other.

• More date security.

**(E) C vs. C++**

C

• C is a middle level language

• C supports function programming

C++

• C++ is a high level language .

• C++ supports oop’s concepts.

**(F) C++ vs. Java**

C++

• It support pointer concept.

• It have structure and union concept.

• C++ use only compiler

Java

• Java doesn’t support pointer concept.

• Java does not include structures or unions.

• Java uses compiler and interpreter both.

**(G) Syntax error vs. Logical error**

Syntax error

• A syntax error happens because of a problem in the program's syntax.

• A syntax mistake can be found more easily.

Logical error

• It is quite challenging to uncover logical errors.

• Logical errors result from algorithmic flaws .

**Answer 3**

01. Virtusa

One of the biggest and most established IT service providers in Sri Lanka is Virtusa, which was established in 1965. Virtusa specializes in offering industry IT solutions to big businesses so they can keep up with the latest technical

advancements and shifting global business trends. Visas offers a broad range of IT Services, including cybersecurity, data and analytics, UX and interface design, intelligent automation, digital process automation, EAP to payment modernisation, mobile game testing services, and Al managed services, among others. Partnered with reputable industry leaders like Adobe, AWS, Google, Page, Salesforce, and others, Vitus provides cutting-edge IT solutions to more than 200 businesses to help them reach their full professional potential and keep up with the rapidly evolving digital world.

02 .Nexdin

With its headquarters in Jalal, Sri Lanka, Nexdin is a recently founded, rapidly expanding internet marketing company in that country. They are experts in game creation, search engine optimization, website design and development, mobile application development, and local business SEO. Nexdin, an IT company with three years of expertise, offers cutting-edge and scalable IT solutions to businesses eager to adapt with the times and expand their reach to new markets.

03.WESO2

Found in 2005, WSO2 is a leading open source integration retailers in SRI Lank. the open-source integration service provided by WSO2 are a swift move towards the future trends of the IT industry and it enables a fast and free flow of data across to potential consumers .API manger, Enterprise integrator , identity saver are some of the popular products expertise by wso2 .with 15 years of experience and a worldwide customer base spanning across healthcare ,finance ,retail, telecoms ,etc.; WSO2 provides software solutions to some renowned enterprises, including eBay, The Warehouse group Karnataka Bank Ltd ,Qantas etc.

04. John Keels IT

John keels IT a subsidiary company of its parenting company; john keels holding PLC, which specialize in technology consulting. Headquartered in Colombo, john keels IT is one of the oldest IT solution companies in Sri Lanka with more than 20years of experience in the industry ,helping motivated enterprises to equip themselves with technology –based business solution and strategies to improve their efficiency and effectiveness .among the most prominent service provided by john keels IT are IT strategy, digital marketing consultancy etc.; with a client –base of over 500client ,john keels takes pride in their innovative trend –setting capabilities.

05. Epic technology

Epic is one of the leading software companies in Sri Lanka situated in Baththaramulla with a long term experience of nearly two decades in the industry. pioneering in digital transformation, e government solution, workflow management, secure electronic payment automation, information system security, and mobile enterprise automation solutions Epic Technology is known for the innovative, ultramodern is active at

An international level especially in southeast Asia, across diverse business verticals. marking territorial excellence in the finance sector, Epic has established a cliental base with almost all the government and private banking companies in Sri Lanka including the central Bank of Sri Lanka, NSB, BOC, people’s Bank HNB, Sylvan Bank, Union Bank etc.

**Tutorial 02**

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

Single line comment - // comment here

Multiple line comment - /\*write comments here\*/

The purpose of adding comments is to make it easier for anyone reading the code or the coder to understand how the code works.

**2.What is the function that is essential in a C program?**

The main() function is essential. This function will run the code written in it.

**3.What is the purpose of ‘scanf’ ?**

The scanf function will receive user input and assign it to the provided variables.

**4.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

(c) file-3 –invalid

(d) return- invalid

(e) $tax – valid but not suitable

(f) name- valid

(g) name and address-invalid

(h) name-and-address-invalid

(i) name\_and\_address- valid

(j) 123 - 45 - 6789- invalid

**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, \n is used the print output in a new line

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

False, text entered between the comment symbols \* \* is only accessible in the code's source code. To make it simple for programmers to understand the objective of the code, comments are employed.

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.

False, Number and NuMbEr are two separate variables in C because of the case-sensitivity of the language.

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

False, Three printf statements are used, but only one line of output is generated.

Where the line has to break, use n to get three lines.

**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, z);

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

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

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.

True

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

True

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

True

**Tutorial 03**

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

x=x+1

printf(“x:%d”,x);

printf(“x+1:%d”,x+1);

printf(“x++:%d”,x++);

printf(“++x:%d”,++x);

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

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

z=x++ + y ;

int x=10, y=5, z =x++ + y;

printf("Output:%d ",z);

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

x\*=2;

int x=10;

printf(“Output:%d”,x\*=2);

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

x=x\*2;

int x=10;

printf(“Output:%d”,x=x\*2);

d)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.);

}

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

total-=--x;

int x=10, total=100;

printf("Output:%d",total-=--x);

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

total+=x—;

int x=10, total=100;

printf("Output:%d",total+=x--);

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

q %= divisor;

q = q % divisor;

int q=10, divisor=3;

printf("Output:%d", q %= divisor);

printf("Output:%d", q = q % divisor);

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

123.45

printf(“%.2f”,123.4567);

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

3.141

printf(“%.3f”,3.14159);

**Q3. Write single C statements that**

a) Input integer variable x with scanf. : scanf(“%d”,&x”);

b) Input integer variable y with scanf. : scanf(“%d”,&y”);

c) Initialize integer variable i to 1. : i=10;

d) Initialize integer variable power to 1. : power=1;

e) Multiply variable power by x and assign the result to power. : power\*=x;

f) Increment variable i by 1. : ++I;

g) Test i to see if it’s less than or equal to y in the condition of a while statement. While( i<=y)

{

}

h) Output integer variable power with printf.

printf(“output:%d”,power);

**Tutorial 04**

**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;

• The second statement of the if condition should have equal signs.

• There shouldn't be any brackets after the if statement.

• "If" and "Else" statements both require curly brackets..

if (numNeighbors >= 3 || numNeighbors == 4 )

{

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

} else

{

--numNeighbors;

}

**2.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");

No, I’m here!

No, actually, I’m here!

**3.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;

**4.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’)

{

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

}

–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)

{

printf(“Sum: %lf”,x+y);

scanf(“%lf %lf”,&x,&y);

}

–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\*=3;

++currentNumber;

}

else

{

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=1;

}

–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 than500 8.00

More than 500 but less than 1,000 10.00

1,000 or more 12.00

If (distance>=0 || distance<=100)

{

cost=5.00;

}

elseif (distance<=500)

{

cost=8.00;

}

elseif (distance<1000)

{

cost=10.00;

} else

{

cost=12.00;

}

**Tutorial 05**

**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 \_\_\_

#include <stdio.h>

#include <stdlib.h>

int main()

{

int n1,n2,choice,sum,sub,mul,div;

printf("Enter two numbers: ");

scanf("%d",&n1);

scanf("%d",&n2);

printf("1.+ \n2.- \n3.\* \n4./ \n");

printf("Enter your choice: ");

scanf("%d",&choice);

sum=n1+n2;

sub=n1-n2;

mul=n1\*n2;

div=n1/n2

switch (choice)

{

case 1:printf("sum = %d",sum);break;

case 2:printf("substraction = %d",sub);break;

case 3:printf("multiplication = %d",mul);break;

case 4:printf("division= %d",div);break;

}

return 0;

}

**While loop**

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

#include <stdio.h>

#include <stdlib.h>

int main()

{

int counter=1,odd=0,even=0,no;

while (counter<=10)

{ printf("Enter %d number: ",counter);

scanf("%d",&no);

if (no%2==0)

{ even=even+1; }

else

{ odd=odd+1; }

counter=counter+1;

}

printf("Total count of odd numbers: %d\n",odd);

printf("Total count of even numbers: %d\n",even);

return 0;

}

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

#include <stdio.h>

#include <stdlib.h>

int main()

{

int counter=1,odd=0,even=0,no;

while (counter<=10,no!=-99)

{

printf("Enter %d number: ",counter);

scanf("%d",&no);

if (no%2==0)

{ even=even+1; }

else

{ odd=odd+1; }

counter=counter+1;

}

printf("Total count of odd numbers: %d \n",odd);

printf("Total count of even numbers: %d \n",even);

return 0;

}

**Do while loop**

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

Q1

#include <stdio.h>

#include <stdlib.h>

int main()

{

int counter=1,odd=0,even=0,no;

do

{

printf("Enter %d number: ",counter);

scanf("%d",&no);

if (no%2==0)

{ even=even+1; }

Else

{ odd=odd+1; }

counter=counter+1;

}

while (counter<=10);

printf("Total count of odd numbers: %d \n",odd);

printf("Total count of even numbers: %d \n",even);

return 0;

}

Q2

#include <stdio.h>

#include <stdlib.h>

int main()

{

int counter=1,odd=0,even=0,no;

do

{

printf("Enter %d number: ",counter);

scanf("%d",&no);

if (no%2==0)

{ even=even+1; }

else

{ odd=odd+1; }

counter=counter+1;

}

while (counter<=10,no!=-99);

printf("Total count of odd numbers: %d \n",odd);

printf("Total count of even numbers: %d \n",even);

return 0;

}

**For loop**

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

#include <stdio.h>

#include <stdlib.h>

int main()

{

int no,sum=0,i;

float avg;

for (i=1;i<=10;i++)

{

printf("Enter the number: ");

scanf("%d",&no);

sum=sum+no;

}

avg=sum/10;

printf("Sum is %d \n",sum);

printf("Average is %.2f ",avg);

return 0;

}

**2. Display the following output using the for loop**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

#include <stdio.h>

#include <stdlib.h>

int main()

{

int i,x,row=5;

for (i=1;i<=row;i++)

{

for (x=1;x<=i;++x)

{

printf("\*");

}

printf("\n");

}

return 0;

}