Programming Concept for ALL
National Examination Board Nepal
Secondary Education Examination

1. Define a program.

A program is a systematic composed of statements, operators, and expressions following certain syntax. It is written in a specific purpose. The nature of a program may differ with another program. It is written by using a programming language. The program written by using a programming language in a human readable form is called Source Code, that is translated into Object Code by the language processor called Assembler or Interpreter or Compiler and again they should be linked with the required library files to be executable independently and a program gets the completion to do any specific task. And the integration of multiple programs would create software.

Q.2. What is programming?

Programming is a process of writing a program by using any computer language. The complete task from the beginning to the end of the program development comes in programming. For this, a

very good skill and knowledge of various programming language and database requires. The person who develops the program is known as a programmer. It also can be of two types, System programming, which is the task of developing system programs and Application Software programming, which develops the application software.

Q.3. What is programming language? Define and write the types of it.

Programming language is a set of symbols, codes and reserved words made to develop or write the program. It helps to develop program to instruct any machine. In this sense, the language used for program writing is known as programming language. It is of basically two types:

a) Low Level Language:

The programming language, which is near to the machine language, is known as Low Level Language. It is also of two types:

i) Machine Language: It is made up of complete binary digits, 0 and 1. It is very hard to use but this one is the real machine language. It is because the machine understands only 0 or 1 or off and on state of electric circuit. And since it is the language of

machine itself, it does not require the translator to understand by the machine. So, it runs very fast in the machine. But disadvantage of it is, difficult to write the program and difficult to debug. Similarly, a program written for one machine does not work in another machine. This case is known by the term 'Machine Dependent'.

ii) Assembly Language: The low level language made up of mnemonic codes is known as Assembly language. The mnemonic codes are the set of reserved words to instruct machine like ADD for addition, SUB for subtraction, etc. After the development of Assembly language, the program writing was quite easier than the machine level language. On the basis of nature of complexity of program writing, Machine Level is known as 1st generation computer language and Assembly language is known as 2nd generation computer language. Though the program writing was easier than Machine level language, still the complexity was existed in the Assembly language because the developer had to know the exact reserved word for the function or instruction. Similarly, it requires the translator called assembler to understand by the machine because it is not the language of machine.

So, the program execution was slower than the Machine language and still it is machine dependent language.

b) High Level Programming Language

The programming languages, which are made up of simple English statements and grammatical syntax are known as High Level Programming language. They are far from the understanding of machine, so, they should be translated into machine codes by using either Interpreter or Complier. These two Interpreter and Compiler are the language processor or translator of the High Level Language into Machine Level Language. Program writing in High Level is quire easier than the earlier two languages Machine level and Assembly Language. But program run is slower and consumes more memory. They are quite popular among the programmers because they are machine independent, it means, same program can be used in all the machines of same types. But it is also of different types as follow:

i) Procedural Oriented Language (3^{rd} Gen. Language)

This type of language requires writing all the details of any function or output. It means the program

should write all the step by step details of any output or function in program. But still the program development is easier than the earlier. Pascal, BASIC, COBOL, C, FORTRAN, etc. are the examples of this type.

ii) Problem-Oriented (4^{th} Gen. Language)

This type of language provides the ready-made buttons, icons, windows and other tools to develop the software, so that the program development gets easier than the earlier. The developer only selects the required tools for the program; it means does not write whole detail of program writing. If he requires window or any icon, just he selects them and does not concern how they are made. Visual Basic, C#, PHP, etc. are the examples of this type.

iii) Natural Language (5^{th} Gen. Language)

These programming languages are still in under development. It has been supposed that they will use simple English statements or other statements which we normally use in our speaking. Then, the program writing will be quite easier than before. But PROLOG (Programming Logic) has been used as a 5^{th} Generation language now. And specially, they will be used in the 5^{th} Generation Computers.

Q.4. Define Language Processors (Translators).

These are the special software developed for the translation of Assembly Language and the High Level Language into the Machine Language. The machine understands and can process only its language i.e. Binary language. So, any instruction written in other human readable language should be translated into the machine readable language. For this purpose, some translators have been developed. They are the language processors or the translators. They are of three types. They are:

a) Assembler: A type of language processor, which translates the source code or program code written in the Assembly Language into the Machine Language or Machine Readable Code, is known as Assembler.

b) Interpreter: It is also a type of language processor, which translates the source code written in High Level into Machine Code. It translates line by line. Until the translation of one line is not finished, it does not proceed for the next line. So, the translation is slow but it gives more error free compilation.

c) Compiler: It is also a translator of High Level into Machine Level like the Interpreter but it translates whole program code at once. So, compiling process

or the translation is faster than the Interpreter but it produces more errors.

Q.5. Define the terms Testing and Debugging.

These both terms are used in the software development. Testing means to test a recently developed software whether that meets the requirement of the clients or not, or the intended result or not. During this process, the program may have many errors called bugs. And the processor of correcting them is known as Debugging. During this process, the source code is compiled first, and the compiler shows the errors if exist and a programmer debugs them accordingly.

Q.6. Define Syntax and Semantics.

Syntax is the grammatical format or set of rules to write any program statements. On the other hand, Semantics are the meaning of each individual words and statements used in the program code.

For example:

int num = 4;

Here, first data type, then variable name and assignment of the value and semicolon in the last are the rule of writing and assigning a value to the variable in c language, which is the syntax. And it cannot be changed. Similarly, int means integer

data type, num means a variable, semicolon is the terminator of a statement, etc. are the semantics.

Q.7. What is the term "Error"? Define and write the types of them.

Errors or the mistakes are the causes of mis-programming or wrong programming done by the programmers. It can be through the mistyping in the syntax and semantics. They are known as the bugs and correcting them is known as debugging. There can be three types of mistakes or errors in a program. They are:

a) Syntax Error: The errors in the grammatical form of the program or commands or in the syntax of the statements are known as the Syntax Error. For example, in a c programming, variable name should be initiated by the data types and simple statements should be terminated by the semicolon. If a programmer forgets writing them, program does not run, this is known as Syntax error. This type of error can be easily found due to the grammatical format. Such syntax of any programming language is easily available in the market to learn. So, a fresh candidate also can debug such errors.

b) Semantic Error (Logical Error)

This is the errors in the words, operators and the expressions used in the program. This would be correct in the syntax but would be mistake in the logical form. For example, in a program to calculate the product of any two number, we may write p=x*y; but if we write p=x+y; it also runs but we get the sum of the entered two numbers, not the product. Such errors in the programming would be very difficult to detect because the compiler software cannot detect them. Only the experienced programmers can detect and make correction them. So, the debugging cost is expensive for them due to the hiring of experienced programmers.

c) Run-time Errors: Even after the correction of a program from the syntactical and semantically errors, program may get error in the run time. It is not actual error of program but while designing or writing program, if the memory consumption of program is not considered where it would be used, then such mistake might be found. For example, the leakage memory, over flow of memory, lack of memory, etc. are the common errors of Run-time errors.

Q.8. What are the data types?

The term data types means, the types of the data to be used in a program in run time. There are different data types. Some of them:

a) Numeric Integers: These are the whole numbers without decimal like 1, 2, 67, -3, +6, etc. But normally, positive integers are not written with + sign. It means, +7 or 7 means same. But negative numbers should be written with -, so they are called signed integers and positive integers are called unsigned.

b) Numeric Real: Numeric Real can be Integers and Fractional numbers both. For example, 67, -23, 12.66, 2.33, etc.

c) Characters: It can be alphabets and any other alpha-numeric characters and symbols, which can be printed. Example, a, z, #, @, a12, b66, etc.

d) Boolean (logical) data type: It is a data type, which can be either True or False values, only.

Q.9. Define Variable and Constants.

Variables are the identifiers or the place holders in the memory identified by a certain name in memory location of machine, which takes the values temporarily in program run time. Similarly, Constants are the values assigned to the variables.

Variables are always written to the left hand side of = sign and the constants are always written to the right hand side of = sign. Variables can be written without assigning values, too. But Constants cannot be written without defining variables. Example:
int num = 4;
Here num is an integer variable and 4 is a constant assigned to variable num.

Q.10. What are the program design tools? Explain them.

The set of tools, which are used for designing a software from the initial state to the end of the software development, is known as program design tools. Algorithm, Flowchart, Pseudo code, Decision Tree, Structured English, Context Diagram, DFD (Data Flow Diagram), etc. are common examples of them.

* Algorithm : The set of step by step procedures to perform any task is known as Algorithm. It does not specify for any particular programming language, rather it suits for any language. So, it should be written in simple and short language to understand. Example of an Algorithm to calculate simple interest can be as follow:

STEP 1: Start
STEP 2: Input P, T, R
STEP 3: Calculate si=(p x t x r)/100
STEP 4: Display si
STEP 5: End
Thus, an algorithm can be of any number of steps but better to write in few steps and clear language.

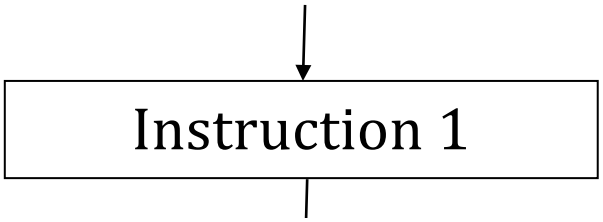
***Pseudo code:** It looks similar to a program but it does not use any specific syntax of program. It does not use the words STEP, too, like in an Algorithm. It uses English phrases and mathematical statements to describe the processing steps of a program.
Example: Write a pseudo code to calculate si.
get p,t and r.
calculate (p x t x r)/100 and store in si
display si
Note: The above statements or phrases do not belong to any language but look like a program.

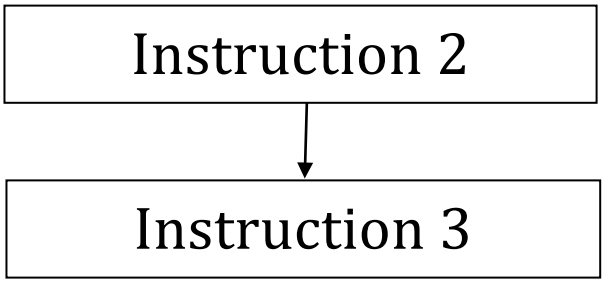
*Flowchart: It is a graphical representation of any algorithm. It uses specific or standard graphical tools or symbols to represent the steps of algorithm. It is of two types:

a) System Flowchart: The flowchart, which gives the outline or outer view of the data flow in whole system, is known as System Flowchart. It means it gives the detail operation of a system.
b) Program Flowchart: It is a part of a system. It does not show over all data flow but it displays the detail operation of a certain part or procedure in a system. It uses the following tools:
Page 194

Q.11. Define the term "Program Control Structure" and write the types of it.

Program Control Structure refers to the flow of program execution in a program while running. It means after execution the program how data and program execution goes. On the basis of program execution flow, there are three types of Program Control Structure. They are:
a) Sequence or sequential Control Structure: In this structure, a program lines execute one after another in serial. It goes from the top to bottom turn by turn. It can be shown as follow:





b) Selection/Conditional/Branching Structure: In this structure, program execution strikes the condition and if the condition is true, operates on first statement otherwise, operates on second statement. So, it uses if condition for this.

Fig.

c) Iteration (Looping/Repetition)

It is used for repeating a certain block of statements until the condition is not over. For example to repeat the PRINT statement numbers from 1 to 10, printing same name 20 times, etc. This is used for loop statements.

Fig.

Q.12. Write the various Data Representation Codes.

Since the computer is a machine and it understands only machine language, which is limited to 0 and 1, everything we give to it, should be translated into certain number and changed into its own language for processing. For this reason, there are various

Data Representation Codes for the letters, numbers, symbols, pictures, etc. Some of them are:

a) Absolute Binary

It is used to represent only the numbers. In it, the whole number is converted into the binary numbers. For example, 4 means 100, 8 means 1000, etc.

b) BCD (Binary Coded Decimal)

It is different from Absolute Binary but easy to convert the numbers. In this process, each number is converted into 4-4 digit or bits of binary digits. For example, 34 means 0011 0100, Similarly 90 means 1001 0000, etc.

c) EBCDIC: It stands for Extended Binary Coded Decimal Interchange Code. It is similar to BCD but it represents with 8-8 bits for each digit.

d) ASCII (American Standard Code for Information Interchange)

It is very popular method of input devices. Most of the input devices follow this method to convert input given to the CPU. It uses some standard number representation for each alphabet and digits like A means 65, B means 66, etc.

e) Unicode: This is also very popular method of data representation these days. It provides up to 4 bytes (32 bits) representation of letters, numbers and

symbols, which means, more than 4 billion different characters and symbols used in computer.