

Computer Programming / - Teacher ICT Retooling Workshop Sept 2017

Program Development Cycle (Stages in Program Development)

1. Problem recognition: This refers to the understanding and interpretation of a particular problem. To understand a problem one has to look for key words such as compute, evaluate, compare etc. A programmer identifies problems in the environment and seeks to solve them by writing computer program that would provide the solution.

Circumstances that can cause the programmer to identify a problem

- a) Opportunity to improve the current program
- b) A new directive given by the management requiring a change in the status quo.
- c) Problems or undesirable solutions that prevent an individual or organization from achieving their purpose.

2) Problem definition: At this stage the programmer tries to determine the likely input, processing and expected output. Several methods to solve a problem are identified and the best one is chosen. At the end of the stage requirements documentation for the new program is written.

3) Program design: This is the actual development of the program's processing or problem solving logic called algorithm. An algorithm is a limited number of logical steps that a program follows in order to solve a problem. The Algorithm can be written using Pseudo code or a Flow Chart

4) Program coding: This is the actual process of converting a design model/ algorithm into its equivalent program. This is done by creating the program using a particular programming language. The end result of this stage is source programs that can be translated into machine-readable form for the computer to execute and solve the target problem.

5) Program Testing and Debugging: After coding the program has to be tested and the errors detected and corrected. Errors may syntax or logical errors. Debugging refers to detection and correction of errors that may exist in the program. Methods of Error Detection include;

- a) dry-run/desk checking: Going through the program while still on paper before entering it in an editor
- b) debugging utilities: In the program editor, running utilities to detect and correct error
- c) use of test data: Carrying out trial runs of the program with different sets of data

6) Implementation: This is the actual delivery and installation of the new program ready for use, creating data files and train people to use the system. The new system will change the way things are done hence it should be reviewed and maintained.

7) Review and maintenance: This stage is important because of the errors that may be encountered after implementation. A program may fail due to poor use, hence proper training and post implementation support of users will reduce chances of having them entering invalid data that crash the program.

8) Program documentation: This is writing of support materials explaining how the program can be used by users, installed by operators or modified by other programmers. All stages of development should be documented in order to help during future modification of the program.

Types of documentation

- a) User oriented documentation: Is a type of documentation that enables the user to learn how to use the program as quickly as possible and with little help from grammar.
- b) Operator oriented documentation Is the type of documentation that is meant for computer operators e.g. Technicians to help them install and maintain the program.

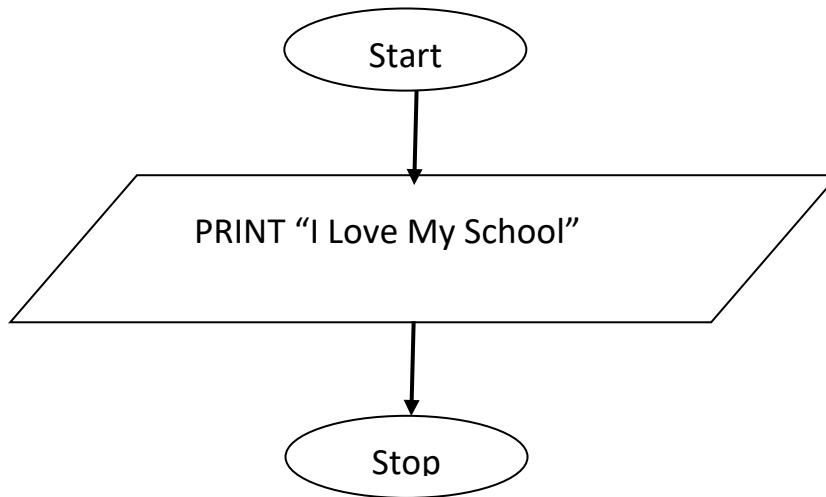
Questions and Answers in Programming Using C Programming Language

Question 1A: Design an algorithm for a program that can output a phrase "I Love My School"

Solution

```
START  
    PRINT "I Love My School"  
STOP
```

Question 1B: Using a Flowchart, design a program that can output a phrase "I Love My School"



Question 1 C: Using C Programming Language, provide a program code that can output a phrase "I Love My School"

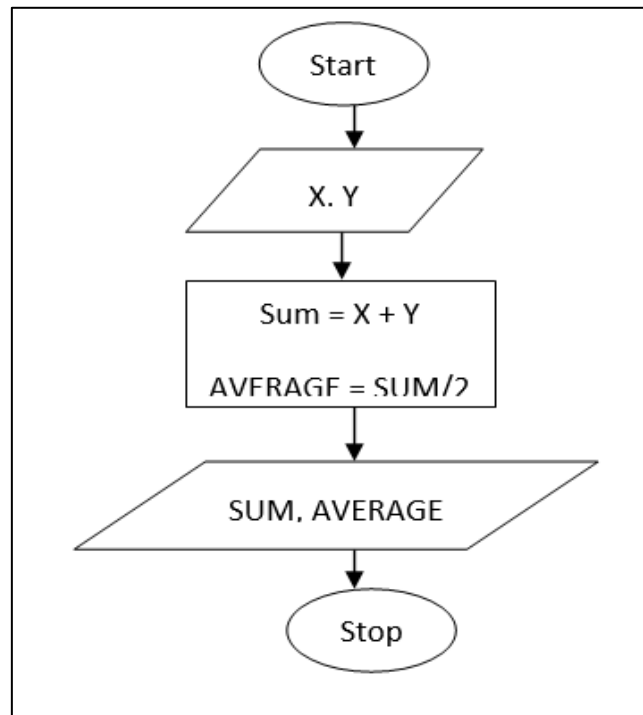
Solution

```
#include <stdio.h>  
int main () {  
    printf("I Love My School\n");  
  
    getch ();  
    return 0;  
}
```

Question#2 A: Design an algorithm for a program that can be used to prompt a user to enter two numbers, calculate the sum and average of the two numbers, and output the sum and average on the screen

Solution#2: Using a Flow Chart

```
Solution#1 USING PSEUDO CODE  
START  
    PRINT "Enter Two Numbers"  
    INPUT X, Y  
    SUM = X + Y  
    AVERAGE = SUM/2  
    PRINT SUM  
    PRINT AVERAGE  
STOP
```



Question#2: Using C Programming Language, write a code that can be used to prompt a user to enter two numbers, calculate the sum and average of the two numbers, and output the sum and average on the screen

Solution

```
#include <stdio.h>  
int main () {  
    int x;  
    int y;  
    int sum;  
    int average;  
  
    printf("Enter Two Numbers");  
    scanf("%d",&x);  
    scanf("%d",&y);  
    sum = (x+y);  
    average = sum/2;  
    printf("The sum is %d",sum);  
    printf("The average is %d", average);  
  
    getch ();  
    return 0;  
}
```

Question#3: Using Pseudocode, design an algorithm for a program that can be used to prompt a user to enter the length and width of a rectangle, calculate the area and perimeter, and output the result on the screen

Solution#1 Using Pseudocode

```
START
    PRINT "Enter length and width":
    READ Length, Width
    AREA = Length* Width
    PERIMETER = 2(Length + Width)
    PRINT AREA
    Print PERIMETER
STOP
```

Question#3: Using either **C Programming Language** or **Flowchart**, develop a program that can be used to prompt a user to enter the length and width of a rectangle, calculate the area and perimeter, and output the result on the screen

Solution#1 Using C Programming Language

```
#include <stdio.h>
int main () {
    int length;
    int width;
    int area;
    int perimeter;

    printf("Enter Length and Width");
    scanf("%d",&length);
    scanf("%d",&width);
    area = length*width;
    perimeter = (length + width) *2;
    printf("The area is %d",area);
    printf("The perimeter is %d",perimeter);

    getch ();
    return 0;
}
```

Solution#2: Using a Flow Chart

