**C Programming Assignment**

**1. Essay: History and Evolution of C Programming**

C programming was developed in 1972 by Dennis Ritchie at Bell Labs. It was created to rewrite the Unix operating system and quickly became popular due to its speed and simplicity. C evolved from earlier languages like B and BCPL. Over the years, C has influenced many other programming languages like C++, Java, and C#. It is still widely used today because it is fast, gives low-level access to memory, and is great for system programming, embedded systems, and learning the basics of programming.

**2. Installing a C Compiler and Setting Up an IDE**

**Steps to install GCC (C compiler):**

1. Go to <https://www.mingw-w64.org/> and download the installer.
2. Install it and set the system path so you can use gcc in the command line.

**Setting up IDEs:**

* **DevC++:** Download from sourceforge.net, install, and it comes with a built-in compiler.
* **VS Code:** Install from code.visualstudio.com. Add C/C++ extension. Set up tasks to compile using GCC.
* **CodeBlocks:** Download from codeblocks.org with MinGW. Install and start coding.

**3. Basic Structure of a C Program**

#include <stdio.h> // header file

int main() { // main function

// this is a comment

int a = 10; // variable declaration

printf("Value of a is %d", a);

return 0;

}

**Explanation:**

* #include <stdio.h>: Header file
* main(): Starting point of the program
* //: Comment line
* int: Data type
* a: Variable name

**4. Types of Operators in C**

* **Arithmetic Operators:** +, -, \*, /, %  
  Example: a + b
* **Relational Operators:** ==, !=, >, <, >=, <=  
  Example: a > b
* **Logical Operators:** &&, ||, !  
  Example: a > 0 && b > 0
* **Assignment Operators:** =, +=, -=, \*=, /=, %=  
  Example: a += 5
* **Increment/Decrement:** ++, --  
  Example: a++, --b
* **Bitwise Operators:** &, |, ^, ~, <<, >>  
  Example: a & b
* **Conditional Operator:** ? :  
  Example: a > b ? a : b

**5. Decision-Making Statements in C**

* **if:**

if (a > 0) {

printf("Positive number");

}

* **if-else:**

if (a > 0) {

printf("Positive");

} else {

printf("Non-positive");

}

* **nested if-else:**

if (a > 0) {

if (a < 100) {

printf("Between 1 and 99");

}

}

* **switch:**

int choice = 1;

switch (choice) {

case 1: printf("One"); break;

case 2: printf("Two"); break;

default: printf("Other");

}

**6. Loops in C**

* **while loop:**

int i = 0;

while (i < 5) {

printf("%d\n", i);

i++;

}

Use when the number of iterations is unknown.

* **for loop:**

for (int i = 0; i < 5; i++) {

printf("%d\n", i);

}

Use when the number of iterations is known.

* **do-while loop:**

int i = 0;

do {

printf("%d\n", i);

i++;

} while (i < 5);

Executes at least once, even if the condition is false.

**7. break, continue, and goto in C**

* **break:** Stops the loop.

for (int i = 0; i < 10; i++) {

if (i == 5) break;

printf("%d\n", i);

}

* **continue:** Skips the current iteration.

for (int i = 0; i < 5; i++) {

if (i == 2) continue;

printf("%d\n", i);

}

* **goto:** Jumps to a label.

int a = 5;

if (a == 5) goto skip;

printf("This won't print");

skip:

printf("Jumped here\n");