**Key words, Identifier, Literals, Operators and Expression Assignment.**

**1. Choose all valid identifiers**

**a. int int**

**b. int \_numvalue**

**c. float price\_money**

**d. char name1234567890123456789012345678901234567890**

**e. char name value**

**f. char $name**

Ans: The Valid Identifiers are:

**int \_numvalue** - Valid. Identifiers can start with an underscore and contain alphanumeric characters.

**float price\_money** - Valid. Identifiers can contain underscores and alphanumeric characters.

**char name1234567890123456789012345678901234567890** - Valid. Although it’s unusually long, it follows the rules for identifiers.

**2. What is the meaning of the following keywords, show the usage.**

**a. auto**

**b. extern**

**c. volatile**

**d. sizeof**

**e. const**

Ans: Auto: The auto keyword is used to declare automatic variables. By default, all local variables are automatic.

Extern: The extern keyword is used to declare a global variable or function in another file.

Volatile**:** The volatile keyword is used to tell the compiler that a variable’s value may change at any time, without any action being taken by the code the compiler finds nearby.

Sizeof: The sizeof operator is used to determine the size, in bytes, of a data type or object.

Const**:** The const keyword is used to declare variables whose value cannot be changed after initialization.

A screenshot of a computer program

Description automatically generated

**3. Explain the difference between the following variables.**

**a. char \*ptr = “ABC”;**

**b. char arr[]=”ABC”;**

a. char \*ptr = "ABC";

* Type: Pointer to a string literal.
* Memory: ptr points to a memory location where the string literal "ABC" is stored. This memory is typically read-only.

b. char arr[] = "ABC";

* Type: Array of characters.
* Memory: arr is an array that contains a copy of the string "ABC" in its own memory space. This memory is writable.

**Can you manipulate the contents of ptr? Why?**

* No, you cannot safely manipulate the contents of ptr because it points to a string literal, which is stored in read-only memory. Attempting to modify it can lead to undefined behavior.

**Can you manipulate the contents of arr? Why?**

* Yes, you can manipulate the contents of arr because it is an array with its own memory space. You can change the characters in the array.

**Which one of the above is a string literal?**

String Literal

* The string literal in this context is "ABC". In the case of char \*ptr = "ABC"; . "ABC" is a string literal stored in read-only memory.

**4. Predict the output of the following code .**

**void main()**

**{**

**//set a and b both equal to 5.**

**int a=5, b=5;**

**//Print them and decrementing each time.**

**//Use postfix mode for a and prefix mode for b.**

**printf("\n%d %d",a--,--b);**

**printf("\n%d %d",b++,--b);**

**}**

A black screen with white text

Description automatically generated

5. Refer the code snippet. It fails with error. Fix it.

#include<stdio.h>

int main()

{

int i,k;

const int num;

/\* for(i = 0;i < 9;i++)

{

k = k + 1;

} \*/

num = num + k; /\* Compiler gives the error here \*/

printf("final value of k:%d\n",k);

printf("value of num:%d\n",num);

return 0;

}

Ans: The error-

A screen shot of a computer error

Description automatically generated

A black background with white text

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

**6. Consider the following code snippet. Evaluate the value of f1, f2 and f3.**

**int main()**

**{**

**int i = 10;**

**int j = 3;**

**float f1 = i / j;**

**float f2 = (float ) i / j;**

**float f3 = (float ) (i / j);**

**}**

Ans: f1= 1/j (both are integers)

1/3 results 0 , f1 is assigned as 0.0 after converting the integer result to float.

**f1=0.0.**

f2: (float)i/j

(float)i-> 10.0

(float)i/j=10.0/3=3.333333.

**f2=3.333333**

f3=(float)(i/j)

(i/j)=(10/3)=3

(float)(i/j)=3.0

**f3=3.0**