# Keywords, Identifier, Literals, Operators and Expression Assignment

**Mandatory:**

1. Choose all valid identifiers
   1. int int
   2. **int \_numvalue**
   3. **float price\_money**
   4. char name1234567890123456789012345678901234567890
   5. char name value
   6. char $name
2. What is the meaning of the following keywords, show the usage?
   1. Auto

* **Auto is a storage class/ keyword in C Programming language which is used to declare a local variable**
  1. Extern
* **“extern” keyword is used to declare and define the external variables.**
  1. Volatile
* **volatile in C came into existence for the purpose of not caching the values of the variable automatically**
  1. sizeof
* **used to compute the size of its operand. It returns the size of a variable**
* #include <stdio.h>
* int main() {
* int a = 16;
* printf("Size of variable a : %d\n",sizeof(a));
* printf("Size of int data type : %d\n",sizeof(int));
* printf("Size of char data type : %d\n",sizeof(char));
* printf("Size of float data type : %d\n",sizeof(float));
* printf("Size of double data type : %d\n",sizeof(double));
* return 0;
* }

1. Explain the difference between the following variables.
   1. char \*ptr = “ABC”;
   2. char arr[]=”ABC”;

**Ans :** char \*ptr = “ABC”;  is  storing the “ABC” as a reference but

char arr[]=”ABC”; Is storing the “ABC” as a value of the Array. We can access those values by arr[0]=”A”,arr[1]=”B”, arr[2]=”C”.

1. Can you manipulate the contents of ptr? Why?

* Yes,A pointer is a variable that holds a memory address of another variable. **The pointer variable can be used to manipulate the contents of the address**. The pointer variable can be used to manipulate the address

1. Can you manipulate the contents of arr? Why?

* Yes,Arrays can be manipulated by using several actions known as methods.

1. Which one of the above is a string literal?

* Array is string literal.

char arr[]=”ABC”;

1. 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);

}

**OUTPUT: a=5, b=4**

**b=3, b=4**

1. 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;

} \*/

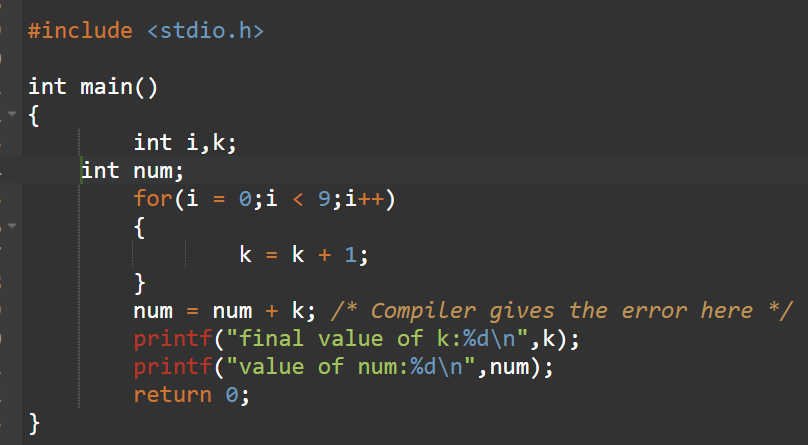
num = num + k; /\* Compiler gives the error here \*/

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

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

return 0;

}

* **we need to remove const from num declaration and also, we need to remove comment keywords to get access to loop**
* 
* 

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);

}

* **Value of f1, f2 and f3 are:**

