# 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**

**Ans:** a. Invalid because keywords cannot be used

b. Valid

c. Valid

d. Invalid because it is too long

e. Invalid as there is space

f. Invalid should not start with $

1. **What is the meaning of the following keywords, show the usage**
   1. **auto**
   2. **extern**
   3. **volatile**
   4. **sizeof**
   5. **const**

A. Auto keyword automatically refers to the type of variable without declaration and allocates memory

b. Extern keyword is used to refer to functions or variables that are defined in another file

c. Volatile keyword describes that the value of variable can be changed at any time

d. sizeof keyword gives the size of datatype or variable

e. const keyword is used for the variables that cannot be altered after initialization.

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

Ans: In a the pointer ptr points to the ABC,in b the array stores the string.

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

1. NO,we cannot manipulate the string but we can manipulate the pointer ptr value to someother value.

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

1. YES,we can manipulate the contents of arr by indexing

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

1. Both the values are string literals.But in a it refers to string literal and in b arr contains string literal.
2. **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);

}

1. 5 4
2. 4 4

**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;

}

A. The error here is a compilation error as it cannot update the value of num as it is constant. By removing the keyword const it works perfectly.

**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 = 3.00

F2 = 3.33

F3 = 3.00