**Keywords, Identifier, Literals, Operators and Expression Assignment**

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

A: a. Invalid because the keywords can’t be used after a identifier.

b. Valid

c. valid

d. Invalid because it is too long

e. Invalid as there is a space

f. Invalid because $ shouldn’t be used

1. What is the meaning of the following keywords, show the usage
   1. auto
   2. extern
   3. volatile
   4. sizeof
   5. const
2. a. auto : it automatically refers to the type of variable without declaration and allocates memory

b. extern: it is used to refer to functions or variables that are defined in another file.

c. volatile: It describes that value of variable can be changed at any time.

d. sizeof: It gives the size of the datatype or variable.

e. const: It is used for the variable that can’t be altered after initialisation.

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

the ptr points to ABC

* 1. char arr[]=”ABC”;

in this the arr stores the string

Can you manipulate the contents of ptr? Why?

A: No we cannot manipulate the contents of ptr/string but we can manipulate the ptr value to someother value.

Can you manipulate the contents of arr? Why?

A: Yes we can manipulate the contents of arr by indexing.

Which one of the above is a string literal?

A:Both are string literals but in this ‘a’ refers to the string literal and ‘b’ in arr contains string literal.

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:

5 4

3 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;

}

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 .

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);

}

OUTPUT:

F1=3.000000

F2= 3.333333 f3=3.000000