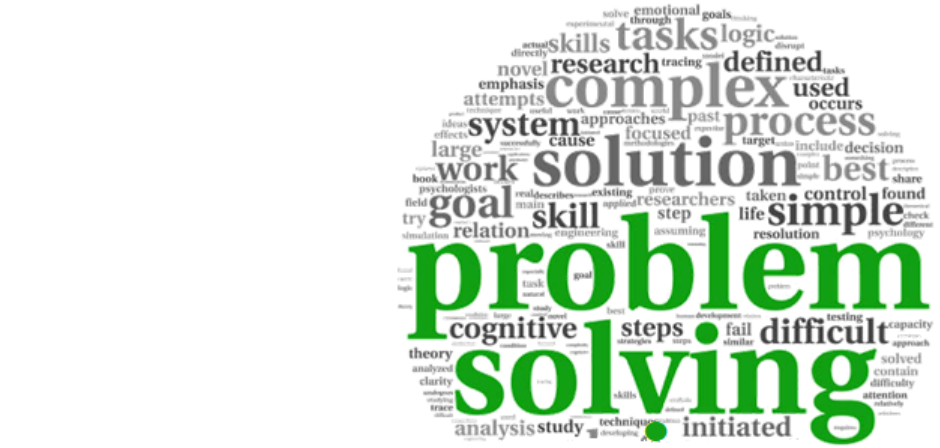
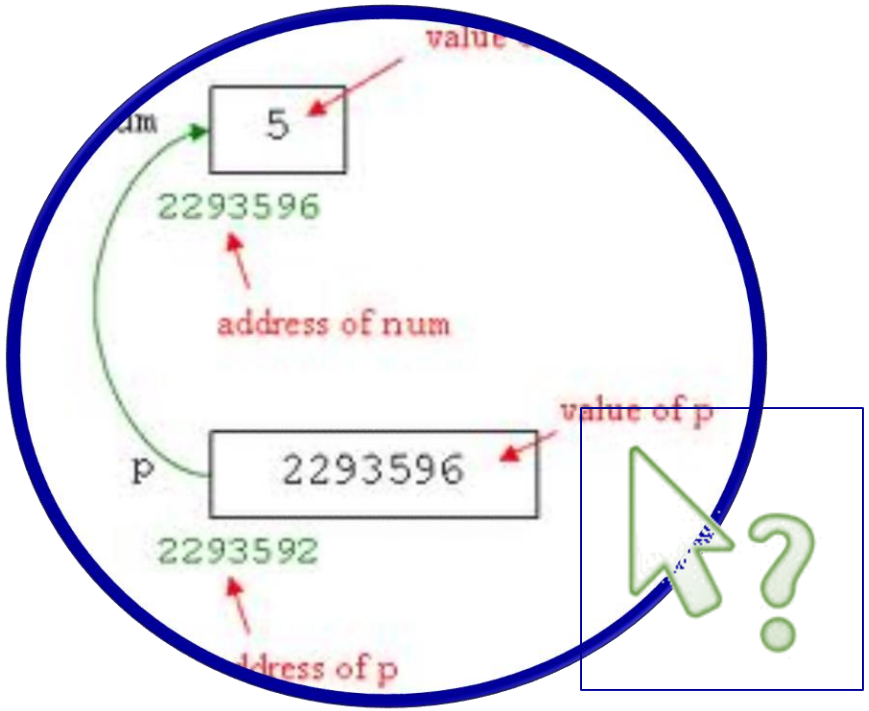

problem
solving
cognitive
steps
difficult
simple
solution
complex
tasks
research
defined
process
system
approaches
goal
skill
relation
work
best
control
researchers
initiated
analysis
study
using
computers

CSE 1051



Pointers S16-2

Objectives

- To learn and appreciate the following concepts:

  - Simple Programs using pointers

Session outcome

- At the end of session one will be able to:
    - Write simple Programs using pointers

Accessing variable through a pointer

- A variable's value can be accessed by its pointer using unary operator *(asterisk) known as indirection operator.

Consider the following statements:

```
int quantity, *p, n;      // 2 int variables & 1 integer pointer
quantity =50;              // assigns value 50 to quantity
p=&quantity;             // assigns the address of quantity to p
n=*p;                      // contains the indirection operator  *
```

* Operator - value at address operator

Example – Accessing variable through a pointer

```
#include <stdio.h>
int main()
{
   int var1 = 11;          //two integer variables
   int var2 = 22;
   int *ptr;               //pointer to integer
   ptr = &var1;            //pointer points to var1
   printf("%d",*ptr);      //print contents of pointer (11)
   ptr = &var2;                    //pointer points to var2
   printf("%d",*ptr);      //print contents of pointer (22)
    return 0;
}
```

Output :
11
22

Example - Accessing via pointers.

```
#include <stdio.h>
int main()
{
   int var1, var2;                    //two integer variables
   int *ptr;                  //pointer to integers
    ptr = &var1;                      //set pointer to address of var1
   *ptr = 37;                 //same as var1=37 ( Dereferencing)
   var2 = *ptr;               //same as var2=var1
   printf("%d", var2);  //verify var2 is 37
    return 0;
}
```

Reference and dereference operators

- & is the ‘reference’ operator and can be read as "address of"
- * is the ‘dereference’ operator and can be read as “value at address” or "value pointed by"

Go to posts/chat box for the link to the question **PQn. S16.2**

submit your solution in next 2 minutes

The session will resume in 3 minutes

Example- understanding pointers

```
#include <stdio.h>
int main()
{
int  firstvalue = 5, secondvalue = 15;
 int * p1, * p2;
 p1 = &firstvalue;       // p1 = address of firstvalue
 p2 = &secondvalue; // p2 = address of secondvalue
 *p1 = 10;       // value pointed by p1 = 10
 *p2 = *p1;      // value pointed by p2 = value pointed by p1
  p1 = p2;                  // p1 = p2 (value of pointer is copied)
 *p1 = 20;       // value pointed by p1 = 20
printf("firstvalue is %d “, firstvalue );
printf( "secondvalue is %d" ,secondvalue);
   return 0;
}
```

Output :
firstvalue is 10
secondvalue is 20

Summary of pointers

- Pointer concept
- Reference operator **&**
- Dereference operator ***