Control flow statements - Part V

Comprehensive Course on C- Programming

Pankaj Sharma • Lesson 12 • May 15, 2024

CS & IT Engineering

C Programming

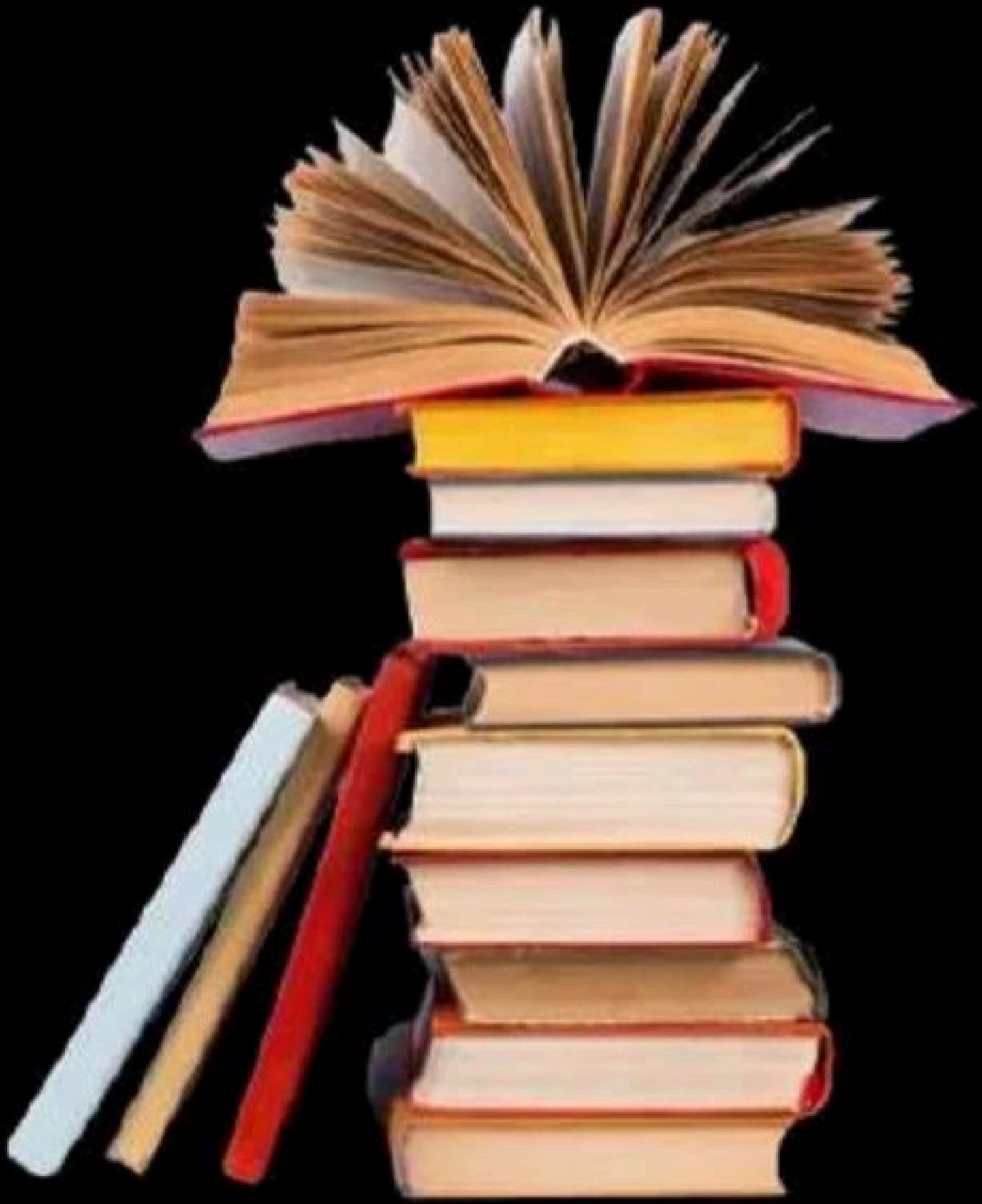
Control Flow Statements-V

Lecture Number- 12

By- Pankaj Sir

Topics

to be covered



1 Switch statement

switch statement

* keyword : used to create selection statement with multiple choices.

* Multiple choices are provided with another keyword

⇒ case

```
switch(n) {

case 1   : Code we want to execute if the
           value of n is 1
           break;
case 2   : Code we want to execute if the
           value of n is 2
           break;
default  : Code we want to execute if the value of
           n does not match any case label.
           break;
}
```

2 x 5 + 3 → Evaluate → 13

```
switch( expression ) {

                1
      case constant1      :   block of statements

                              break;
                 13
      case constant2  →  :   block of statements   ✓

                              break;
                 2
      case constant3      :   block of statements

                              break;
           :
      default             :   block of statements
                              break;
               }
```

```
int i = 3;
switch(i) {               3

    case 1 :          printf("one");
                      break;

    case 3 :          printf("Three");
                      break;

    default :         printf("wrong");
                      break;
            }
```

1st step:

Matching

Sequential

```
int i = 3;
switch(i) {          3

    case 1 :         printf("one");
                     break;

    case 3 :         printf("Three");
                     break;

    default :        printf("wrong");
                     break;
    }
```

1st step: Matching

Sequential

Three

2

Falling through case

```
int i = 3;
switch(i) {
    case 3  : printf("Three"); ✓
    case 4  : printf("Four"); ✓
              break;
    default : printf("zero");
              break;
}
```

3

1st step

3. switch (expression) { → Integer value (constant)

printf("Ankita")

}

Expression
→ 12
→ 1+2x3
→ 12 & 4
→ 12 && 4
→ printf("pankaj")

int a=1,b=2;
→ a+bx3

switch(12.5){

}

vd fre laat Pdegi

'A' ⇒ 65

① Switch('A'){ = ✓

}

Switch('A'+2){ → 65+2 → integer

=

}

```
switch(A)
{

}
```

$\xrightarrow{\quad}$ ud Re laat

'A'+2

③

```
int i = 3;     → 6                         0
switch(i+3) {
      case 5× :    printf("5");
                   break;

      default✓ :   printf("0");
                   break;
       case 7× :    printf("7");
                    break;
         }
```

```
int i = 3;                   ⑥
switch(i+3) {                                              O/P : 07
              ×
    case 5        :     printf("5");
                        break;
            ✓
    default       :     printf("0");  ✓

            ×
    case 7        :     printf("7");  ✓
                        break;

          }
```

```
if (!3)
    pf("3");
else if (!4)
    pf("4");
```

O/P :

Else is optional

```
int i = 3;
switch(i+3){

    case 1 : printf("1");
             break;

    case 2 : printf("2");
             break;

}
```

6

No o/p

default is optional

```
int i = 3;
switch(i+3) {

}
```

valid

dummy

```
int i=3;
switch(i+3);
```

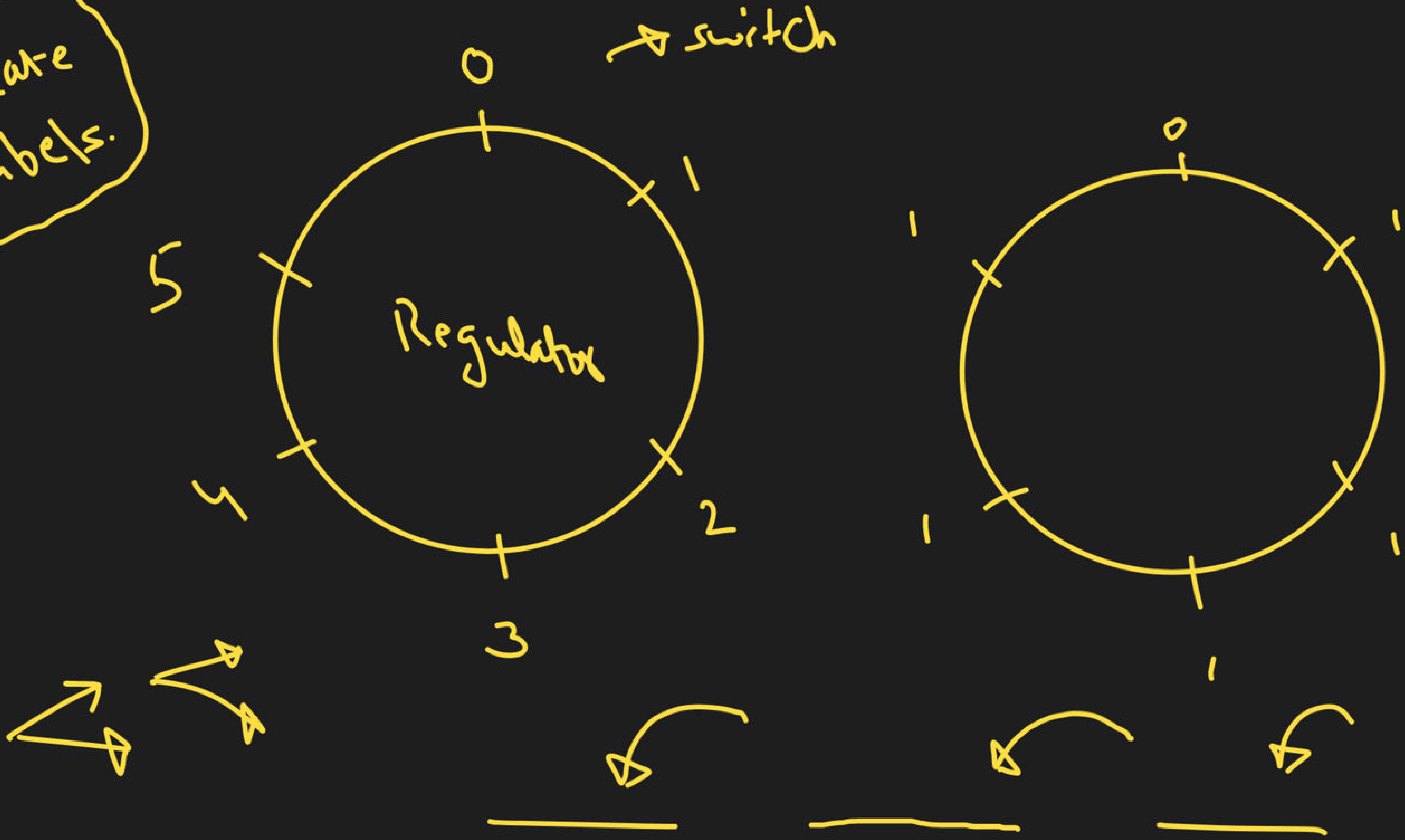
✓

```
switch();
```

ud ke laat Padegi

Must be some exp.

No duplicate case labels.



```
switch(i) {

    case 4 :   □
               break;

    case 2+2 :   □
                 break;

}
```

Error (ud ke laat Padegi)

```
int i=3, a=10, b=30;
switch(i+3){
    case a :  □
              break;

    case b :  □
              break;

    case a+b :  □
              break;
}
```

case labels

must be

=> constant/literals

case 10 ✓

case 10+3 ✓

case 10x3+4 ✓

case printf()

Student → Bahubali

~~Break~~ → Purpose

```
switch(i+3){

    case 1    :  printf("1");
                 printf("2");
                 break;

    case 2    :  printf("3");
                 printf("4");
                 break;

    }
```

code

code

```
Switch(i+3){

    Case 1 :   printf("1");
               printf("One");
               break;
             printf("Pankaj");
    Case 2 :   printf("2");
               break;

             }
```

→ Never gets printed (ignore)

```
int i = 2;

switch(i) {
        i = i + 2;              → ignored

    case 2 :  printf("2");      ✓
              break;

    case 4 :  printf("4");
              break;

    }
```

```
if (n==1||n==13||n==16)
        {

        }
```

Set of values

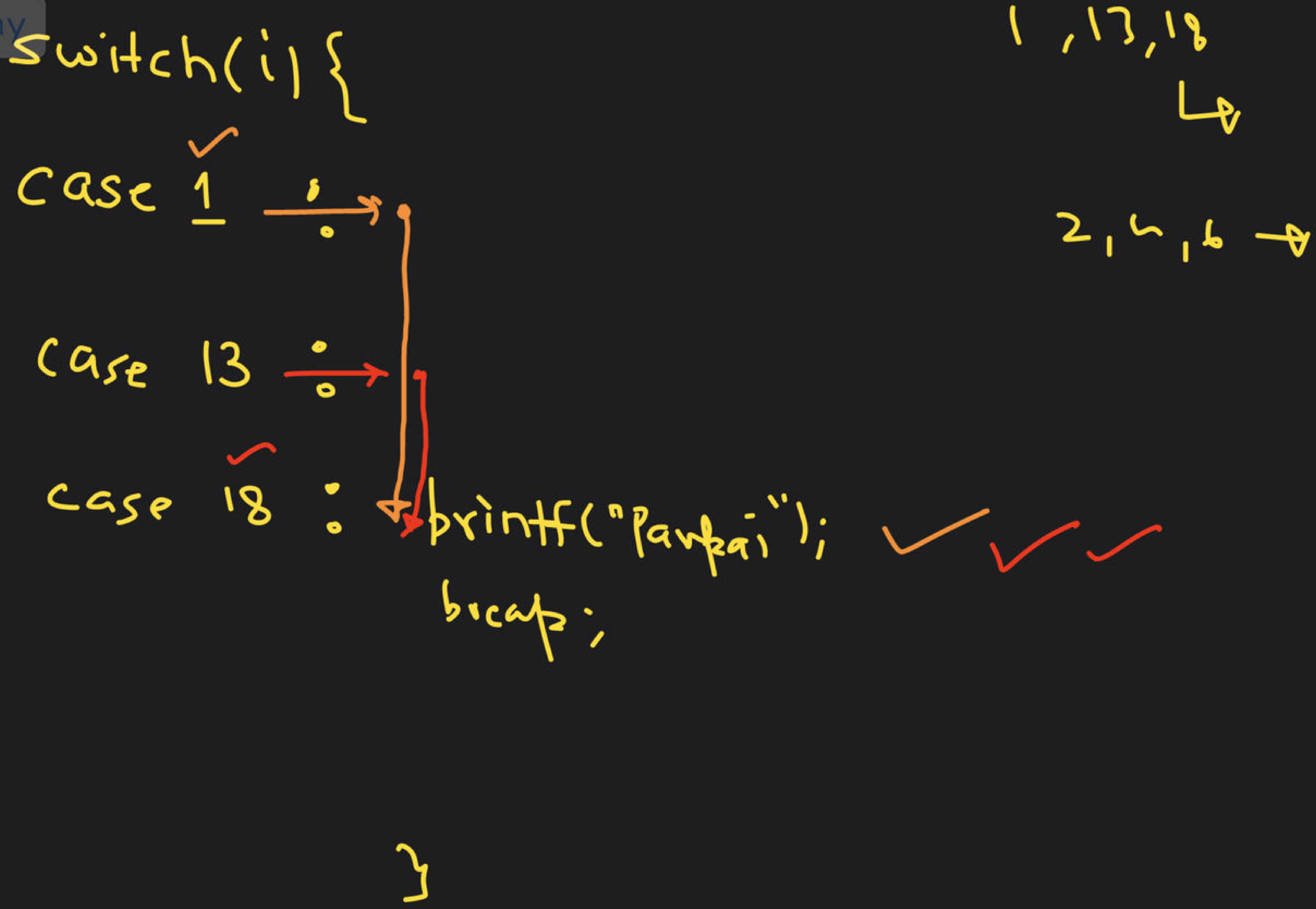
code

```
else if (n==2||n==10||n==15)
          {

          }
```

Set of values

code



```
switch(i) {

case 1 :

case 13 :

case 18 : printf("Pankaj");
          break;

}
```

1 ,13,18 →

2,4,6 →

Not on all compiler

Range

```
if ( i>=1 && i<=10)
{
  —
} ✓
```

```
switch(i){
  case low ... high;  —
}
```

$\text{low}\ \uparrow \ldots \uparrow\ \text{high}$

1space 1space

Not for all Compiler

```
switch(i) {

    case 1...10: printf("Hi");
                 break;
    case 11...20: printf("Bye");
                  break;
}
```

① break is optional.

② expression → evaluates to be integer.

③ position of default does not matter, it can be anywhere

→ start
→ mid
→ End

④ default is optional.

⑤ Duplicate case labels not allowed.

⑥ case labels can only contains constant/ literals

unacademy

THANK YOU!

Here’s to a cracking journey ahead!