

The background features several large, overlapping, semi-transparent swirls in shades of purple, green, and blue. Scattered throughout are numerous small, yellow, triangular shapes that resemble confetti or starbursts.

C Programming Introduction

Week 6: Branches statement



Topic of this week

- **Branches**

- Class Lecture Review

- If selection structure.

- Switch selection structure.

- Programming Exercises



The `if` Selection Structure

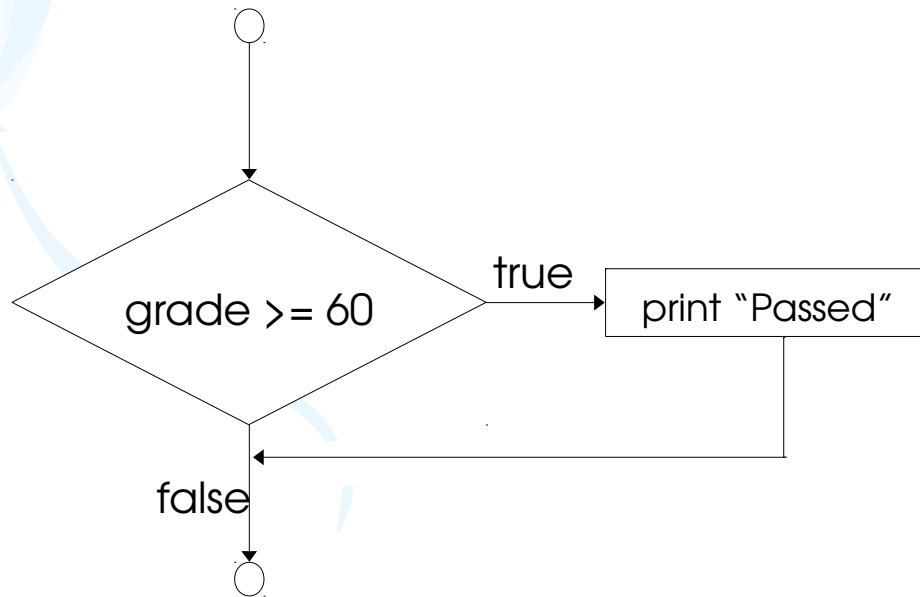
- Selection structure:
 - Used to choose among alternative courses of action
 - Pseudocode: *If student's grade is greater than or equal to 60
Print "Passed"*
- If condition **true**
 - Print statement executed and program goes on to next statement.
 - If **false**, print statement is ignored and the program goes onto the next statement.
 - Indenting makes programs easier to read
 - C ignores whitespace characters.
- Pseudocode statement in C:

```
if ( grade >= 60 )  
    printf( "Passed\n" );
```

 - C code corresponds closely to the pseudocode

The `if` Selection Structure (II)

- Diamond symbol (decision symbol) - indicates decision is to be made
 - Contains an expression that can be **true** or **false**
 - Test the condition, follow appropriate path
- **if** structure is a single-entry/single-exit structure.



A decision can be made on any expression.

zero - **false**

nonzero - **true**

Example:

3 - 4 is **true**

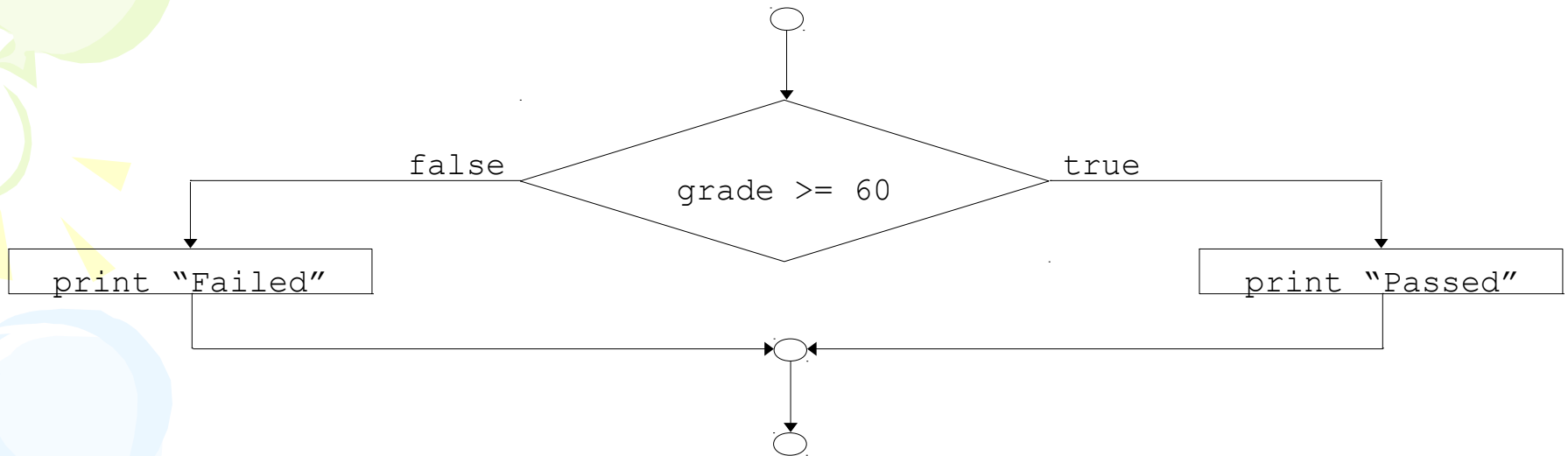
The `if/else` Selection Structure

- `if`
 - Only performs an action if the condition is `true`.
- `if/else`
 - A different action when condition is `true` than when condition is `false`
- Pseudocode: *If student's grade is greater than or equal to 60
Print "Passed"
else
Print "Failed"*
 - Note spacing/indentation conventions

- C code:

```
if ( grade >= 60 )  
    printf( "Passed\n" );  
else  
    printf( "Failed\n" );
```

The if/else Selection Structure (II)



- Ternary conditional operator (?:)
 - Takes three arguments (condition, value if **true**, value if **false**)
 - Our pseudocode could be written:

```
printf( "%s\n", grade >= 60 ? "Passed" : "Failed" );
```

OR

```
grade >= 60 ? printf( "Passed\n" ) : printf( "Failed\n" );
```

The `if/else` Selection Structure (III)

- Nested `if/else` structures
 - Test for multiple cases by placing `if/else` selection structures inside `if/else` selection structures

If student's grade is greater than or equal to 90

Print "A"

else

If student's grade is greater than or equal to 80

Print "B"

else

If student's grade is greater than or equal to 70

Print "C"

else

If student's grade is greater than or equal to 60

Print "D"

else

Print "F"

- Once condition is met, rest of statements skipped
- Deep indentation usually not used in practice

The `if/else` Selection Structure (IV)

- Compound statement:
 - Set of statements within a pair of braces
 - Example:

```
if ( grade >= 60 )  
    printf( "Passed.\n" );  
else {  
    printf( "Failed.\n" );  
    printf( "You must take this course again.\n" );  
}
```

- Without the braces,

```
printf( "You must take this course again.\n" );
```

would be automatically executed

- Block: compound statements with declarations

The `if/else` Selection Structure (v)

- Syntax errors
 - Caught by compiler
- Logic errors:
 - Have their effect at execution time
 - Non-fatal: program runs, but has incorrect output
 - Fatal: program exits prematurely

The switch Multiple-Selection Structure

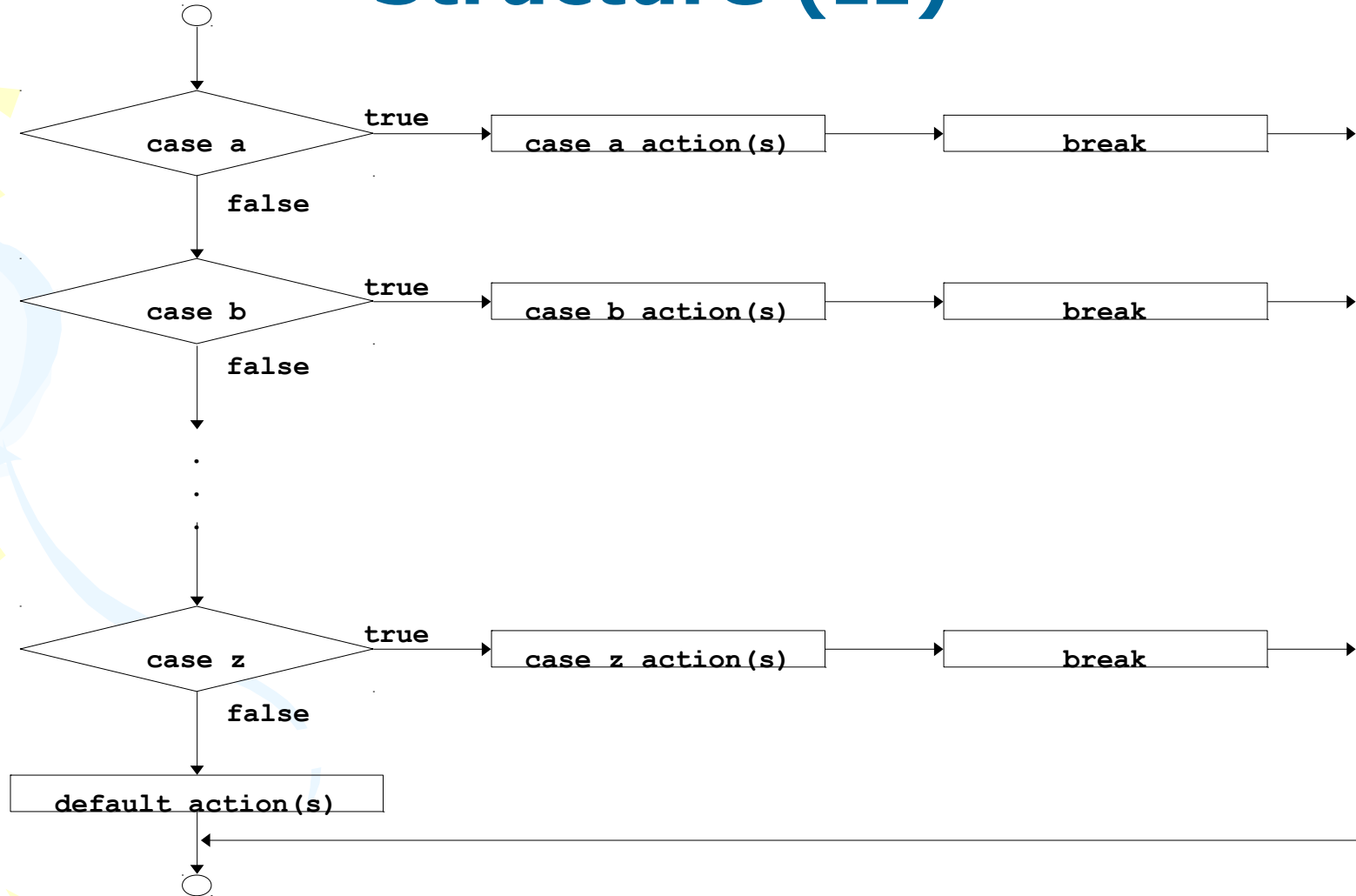
- **switch**
 - Useful when a variable or expression is tested for all the values it can assume and different actions are taken.

- **Format**
 - Series of **case** labels and an optional **default** case

```
switch ( value ){  
    case '1':  
        actions  
    case '2':  
        actions  
    default:  
        actions  
}
```

- **break;** causes exit from structure

The switch Multiple-Selection Structure (II)



The switch Multiple-Selection Structure (III)

- Example of **Switch**

```
c = getchar();
switch (c) {
    case '0': printf("Zero\n"); break;
    case '1': case '2': case '3': case '4':
    case '5': case '6': case '7': case '8':
    case '9': printf("Nine\n"); break;
    case ' ':
    case '\n': newln++; break;
    case '\t': tabs++; break;
    default: printf("missing char\n"); break;
}
```



Exercise 6.1

- Write a program that finds and displays the alphabetically first letter in a sequence (e.g. type IBK and it returns B).
- Note the use of a normal if/else and then an if on its own.



Solution

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char ch1, ch2, ch3; /* three letters (all uppercase or all lowercase) */
```

```
    char alpha_first; /* alphabetically first letter */
```

```
    /* Get three letters. */
```

```
    printf("Enter three uppercase or three lowercase letters> ");
```

```
    scanf("%c%c%c", &ch1, &ch2, &ch3);
```

Solution

```
/* Save the alphabetically first of ch1 and ch2 in alpha_first.      */
   if (ch1 < ch2)
       alpha_first = ch1; /* ch1 comes before ch2 */
   else
       alpha_first = ch2; /* ch2 comes before ch1 */

/* Save the alphabetically first of ch3 and alpha_first.* /
   if (ch3 < alpha_first)
       alpha_first = ch3; /* ch3 comes before alpha_first */

/* Display result.                                                    */
printf("%c is the first letter alphabetically.\n", alpha_first);

return (0);
}
```

Exercise 6.2

- Write a program that transforms a compass heading to a compass bearing using this table:

HEADING IN DEGREES	BEARING COMPUTATION
0 - 89.999...	north (heading) east
90 - 179.999...	south (180.0 - heading) east
180 - 269.999...	south (heading - 180.0) west
270 - 360	north (360.0 - heading) west

- The compass heading is entered by user. Use if/else..if structure.

Solution

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    double heading; /* compass heading in degrees */
```

```
    /* Get compass heading. */
```

```
    printf("Enter a compass heading (0.0 - 360.0 degrees)> ");
```

```
    scanf("%lf", &heading);
```

```
    /* Display equivalent compass bearing. */
```

```
    if (heading < 0.0)
```

```
        printf("Error--negative heading (0.1f): must be between 0 and 360\n",  
heading);
```

```
    else if (heading < 90.0)
```

```
        printf("The bearing is north 0.1f degrees east\n", heading);
```

Solution

```
else if (heading < 180.0)
    printf("The bearing is south %.1f degrees east\n", 180.0 -
heading);
else if (heading < 270.0)
    printf("The bearing is south %.1f degrees west\n", heading -
180.0);
else if (heading <= 360.0)
    printf("The bearing is north %.1f degrees west\n", 360.0 -
heading);
else
    printf("Error--heading > 360 (%.1f): must be between 0 and
360\n",
        heading);

return (0);
```

}



Exercise 6.3

- Write a program that requires you enter an age and shows you what is your class. (child, Senior Citizen or adult)
 - Child : age <18
 - Adult : $18 \leq \text{age} < 65$
 - Senior Citizen: age ≥ 65
- This program should use the if/else structure as a building block in a more complicated structure. It works out a category for people based on their age. Note the way that the if/else structure actually contains another if/else structure.

Solution

```
#include <stdio.h>
int main()
{
    int age;
    /* Get age from user */
    printf("\nEnter an age : ");
    scanf("%d",&age);
    /* Figure out which bracket they are in */
    if(age > 0)
    {
        printf("\nIt is classed as : ");
        if(age < 18)
```



Solution

```
printf("\nA Child\n");  
else if(age >= 65)  
    printf("\nA Senior Citizen\n");  
else  
    printf("\nAn Adult\n");  
}  
else  
    printf("That's not a valid age");  
  
return;  
}
```



Exercise 6.4

- Modify exercise 6.3 by using If/Else structure with conditionals.
- This is a variation on the age program above, see if you can understand the use of conditionals used. You must always be careful about ranges when using the conditionals. What happens if you enter 18 as your age and how can we solve this problem?

Solution

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int age;
```

```
        /* Get age from user
```

```
*/
```

```
    printf("\nEnter an age : ");
```

```
    scanf("%d",&age);
```

```
        /* Figure out which bracket they are in
```

```
*/
```

```
    if(age <= 0 || age >= 125)
```

```
    {
```

```
        printf("That's not a valid age");
```

```
    }
```

Solution

```
else
```

```
{
```

```
printf("\nIt is classed as : ");
```

```
if(age > 0 && age < 18) printf("A Child\n");
```

```
else if(age > 18 && age < 65) printf("An Adult\n");
```

```
else printf("A Senior Citizen\n");
```

```
}
```

```
return (0);
```

```
}
```


Exercise 6.5

- Write a program to play "High/Low". The program "picks" a number. The human player tries to guess it. The program indicates if the guess is too high, too low, or correct. Then it stops.
- Sample outputs:

Results

```
Guess my number (between 1 and 10): 5  
Your guess was too small.  
The correct number was 6.
```

- Use `rand()` function to pick a random number.

Solution

```
/* Guess my number . */
# include <stdio .h>
# include <stdlib .h>
# include <time .h>
int main () {
    int myNum , yourNum ;
    srand ( time ( NULL )); /* Seed */
    myNum = 1 + rand () % 10; /* Pick a number */
    printf ( " Guess my number ( between 1 and 10): ");
    scanf ("%d", & yourNum );
    if ( yourNum < myNum ) {
        printf ( " Your guess was too small .\n");
    } else if ( yourNum > myNum ) {
        printf ( " Your guess was too high .\n");
    } else {
        printf ( " You got it !\n");
        return 0;
    }
    printf ( "The correct number was %d.\n",myNum );
    return 0;
}
```



Exercise 6.6

- Write a program that reads in three integers. and then determine which one is the smallest, and display it.
- If the values are a , b , and c , there are four cases:
 - a is smallest if $a < b$ and $a < c$
 - b is smallest if $b < a$ and $b < c$
 - c is smallest if $c < a$ and $c < b$
 - No smallest When?

Solution

```
/* Least of three numbers */
#include <stdio .h>
int main () {
    int a, b, c, least ;
    printf (" Enter 3 integers : ");
    scanf ("%d , %d , %d", &a, &b, &c);
    if (a < b && a < c) {
        least = a;
    } else if (b < a && b < c) {
        least = b;
    } else if (c < b && c < a) {
        least = c;
    } else {
        printf (" Oops ! No smallest !\n");
        return 0;
    }
    printf (" Smallest is %d\n", least );
    return 0;
}
```



Exercise 6.7

- Alter the exercise6.3 by using Switch selection structure.
- Alter the exercise6.5 by using Switch selection structure.



Exercise 6.8

- Write a C program that does the following:
 - reads the type of a vehicle exiting a car park (C for car, B for bus and T for truck) and the number of hours spent in the car park.
 - calculates the parking fee given the following rates (GST included):
 - Car: \$0.70/hr for the first 2 hours; \$2.50/hr after 2 hours
 - Bus: \$1.50/hr for the first 2 hours; \$2.00/hr after 2 hours
 - Truck: \$2.50/hr for the first hour; \$3.25/hr after 1 hour
 - prints a request for payment that states the total parking fee owed.

Solution

```
#include <stdio.h>
```

```
int main ()
```

```
{
```

```
    char vtype;
```

```
    int hours;
```

```
    float fee;
```

```
    printf("Enter the type of vehicle (C, B, T): ");
```

```
    scanf(" %c",&vtype);
```

```
    printf("Enter the number of hours spent in the car park: ");
```

```
    scanf(" %d",&hours);
```

```
    if (vtype == 'C')
```

```
    {
```

```
        if (hours > 2)
```

```
        {
```

```
            fee = 1.4 + (hours-2)*2.5;
```

```
        }
```

```
        else
```

```
        {
```

```
            fee = hours*0.7;
```

```
        }
```

```
    }
```

Solution

```
// continuing from previous page
else if (vtype == 'B')
{
    if (hours > 2)
    {
        fee = 3 + (hours-2)*2;
    }
    else
    {
        fee = hours*1.5;
    }
}
else // vtype == 'T'
{
    fee = 2.5;
    if (hours > 1)
    {
        fee += (hours-1)*3.25;
    }
}
// note conversion specifier for 2 decimal digits
printf("Please pay %.2f\n",fee);
return 0;
}
```