Revision

Lecture 13

Prepared by: Dr. Yasmine Afify

Final Exam

- Covers all lectures and C++ labs
- Duration: 3 hours
- Grades: 105 for general group

  50 for CH groups

Exam Questions

- Complete
- MCQ
- Compare/Differentiate between
- Convert
- True/False
- Match
- Correct and justify
- Trace
- Display output
- Draw flowchart
- Write a C++ program

Match

Column A	Column B
1. C++	A. code that is easy to understand and to maintain
2. Compiler	B. information hiding
3. Flowchart	C. enclosing many statements between brackets
4. Preprocessor	D. translates code one line at a time
5. Clean code	E. vocabulary and rules of a programming language
6. Loader	F. links object code with the libraries
7. Main ()	G. web-based language
8. Encapsulation	H. combines symbols and flow lines to show algorithm steps
9. Syntax errors	I. translates source code into object code and stores it on disk
10. Syntax	J. entry point of the program execution
	K. processes text which include other files or define constants
	L. procedural language
	M. puts program in memory
	N. detected at compile time

Match: Solution

1	L	Procedural language
2	I	Translates source code
3	H	Combines symbols
4	K	Processes text
5	A	Code that is easy
6	M	Puts program in memory
7	J	Entry point of the program
8	B	Information hiding
9	N	Detected at compile time
10	E	Vocabulary and rules

Display Output

```
int  x = 11, y = 34, number = 12;
while (x < 14  &&  y != 22)
{
(number < 10) ?  y--  :  x *= 2 ;
cout <<  --number  <<  '\t' <<  x++
                       <<  '\t' <<  y/2;
}
```

--number	x++	y/2
11	22	17

Display Output

```
int x = 10, y = 2;
while (x <= 17)
{
cout << x << '\t' <<  y << endl;
x  +=  ++y;
if (x > 11)
continue;
y--;
}
cout << x << '\t' << y << endl;
```

x	y
10	2
13	3
17	4
22	5

Display Output

```
for (int x = 1;  x <= 10;  x += 2)
{
  if (x > 3  &&  x < 8)
  continue;
  cout << x << ',';
}
```

Solution:

1,3,9,

Display Output

```
for (int i = 1 ;  i < 4 ;  i++)
{
   for (int j = i ;  j < 4 ;  j++)
   cout <<  j << '\t';
   cout << '\n';
 }
```

Solution:

1 2 3

2 3

3

Display Output

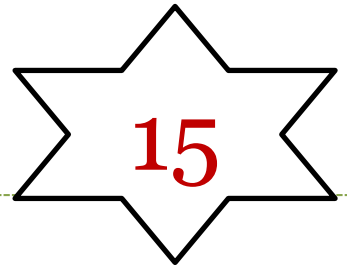
```
int x = 1 + rand() %7;
cout << x << endl;
```

Solution:

Value of random function ranges from 0 to 6

So value of x ranges from 1 to 7

Display Output




```
int sum = 0, n= 3, i = 0;
while (i <= n)
{
      sum += pow (2, i);
      i++;
}
cout << sum << endl;
```

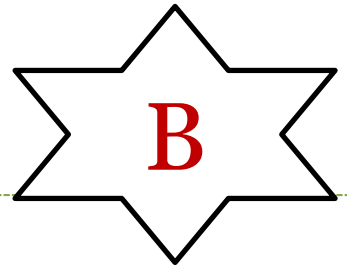
Display Output

```
int x = 2,  y = 6,  z = 3;
x  +=  ++y  *  z++;
cout << x << endl << y << endl << z;
```

Solution:

```
23
 7
 4
```

MCQ

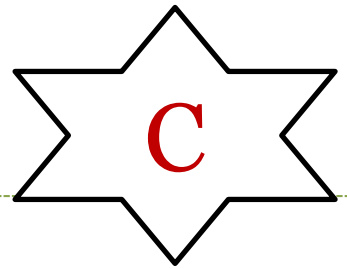



```
bool s = 37 % 2 == 0;

if (s)

cout << "Great ";

cout << "News";
```

This code displays

a) Great News
b) News
c) Great
d) Nothing

MCQ



int x, y, z;

y = 10; z = 15;

x = ++y + z++;

what is the values of x, y and z after executing these statements?

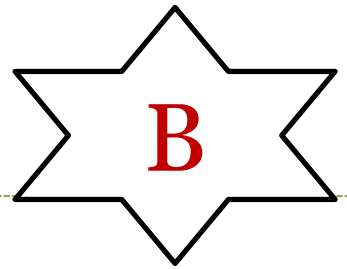
a) x = 25, y = 10, z = 15

b) x = 26, y = 11, z = 15

c) x = 26, y = 11, z = 16

d) x = 27, y = 11, z = 16

MCQ




```
int x = 7;
while (x <= 10)
        cout << ++x  <<  ' ';
```

This code displays

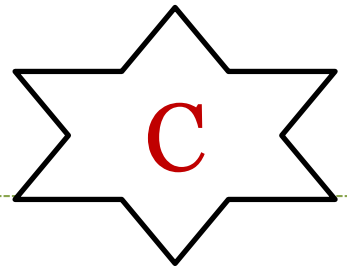
a) 8 9 10

b) 8 9 10 11

c) 7 8 9 10

d) 7 8 9 10 11

MCQ



Which of the following is against the style guidelines for clean code?

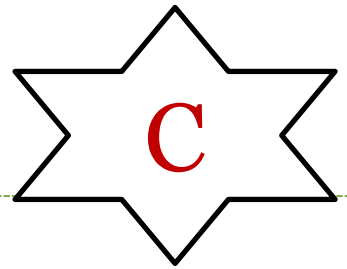
a) Use indentation

b) Negated Boolean variable names should be avoided

c) Abbreviations simplify the code

d) Assign intuitive names to variables

MCQ



The continue statement causes execution to go to:

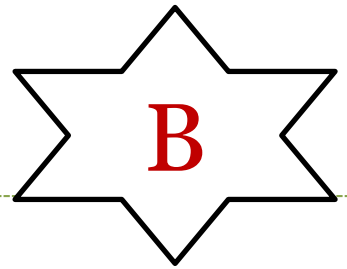
a) return 0;

b) the first statement after loop

c) the following loop iteration

d) the following statement

MCQ



for (; ;)

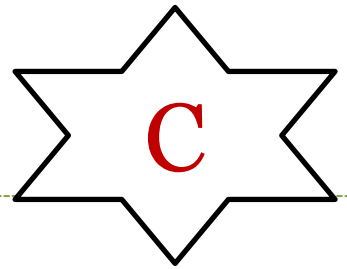
a) similar to while ()

b) similar to while (true)

c) causes syntax error

d) b and c

MCQ



Which of the following generates random numbers from 1 to 14?

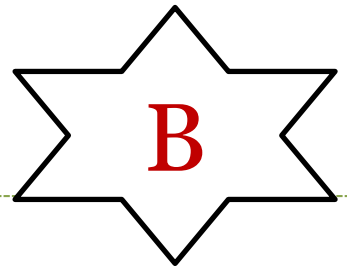
a) 14 + Rand()

b) Rand() % 14

c) 1 + Rand() % 14

d) 1 + Rand() % 15

MCQ



Which of the following cannot be used as an identifier?

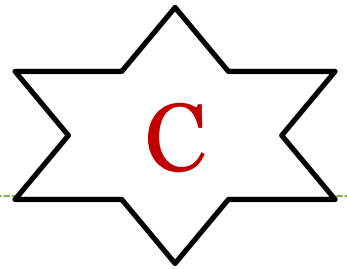
a) _myValue

b) 2scores

c) Price10Products

d) xYz

MCQ



Which of the following is not a syntax error

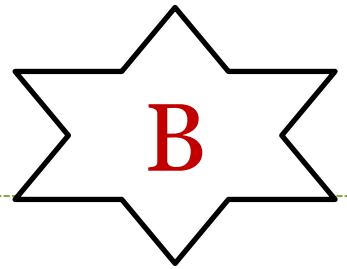
a) Missing semicolon

b) Misspelled identifiers

c) Testing using = instead of ==

d) Forgetting a closing bracket

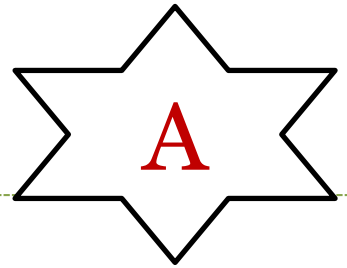
MCQ



Which loop type checks the condition at the end of the loop?

a) For
b) Do while
c) While
d) None of them

MCQ




```
for (int lines = 4; lines > 1; lines--)
{
        for (int s = 1; s < lines; s++)
        cout << '*';
        cout << endl;
}
```

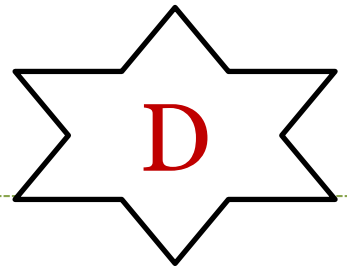
a)
```
***
**
*
```

b)
```
****
***
**
*
```

c)
```
*
*
*
*
```

d)
```
*
**
***
****
```

MCQ



A program instruction that repeats a statement or sequence of statements a number of times

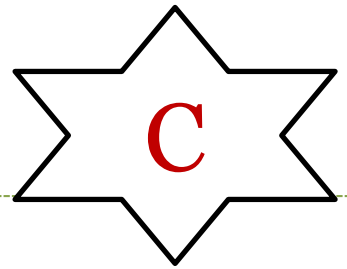
a) Logical operator

b) Escape character

c) Selection Structure

d) Loop

MCQ




```
int sum = 0, n;
cin >> n;
for (int i = 0; i <= n; i++)
        sum += pow (2, i);
```

This statement calculates the series:

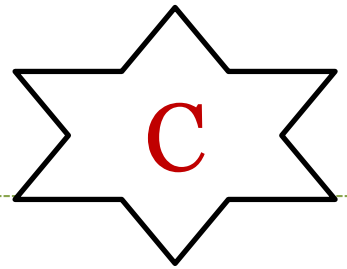
a) $1 + 2 + 2^2 + 3^2 + 4^2 + \ldots + n^2$

b) $2 + 2^2 + 3^2 + 4^2 + \ldots + n^2$

c) $1 + 2 + 2^2 + 2^3 + 2^4 + \ldots + 2^n$

d) $2 + 2^2 + 2^3 + 2^4 + \ldots + 2^n$

MCQ



In a loop, when the condition is always true, you will end up with

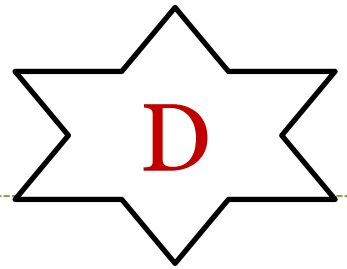
a) Run time error

b) Syntax error

c) Infinite loop

d) Same output for all iterations

MCQ



Which of the following is a run time error?

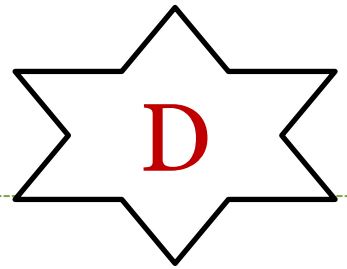
a) Forgetting a header file

b) Undeclared identifier

c) Integer division

d) Division by zero

MCQ



Break statement causes immediate exit from:

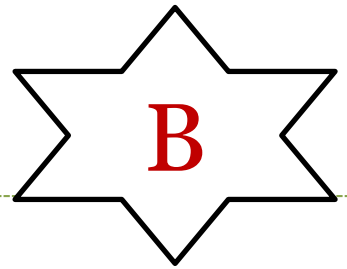
a) loops

b) if statement

c) switch statement

d) a and c

MCQ



int x, y, z;

y = 5; z = 12;

x = y++ + ++z;

what is the values of x, y and z after executing these statements?

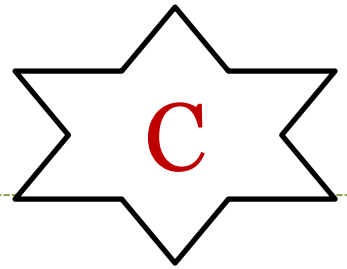
a) x = 17, y = 5, z = 12

b) x = 18, y = 6, z = 13

c) x = 18, y = 5, z = 12

d) x = 17, y = 6, z = 13

MCQ



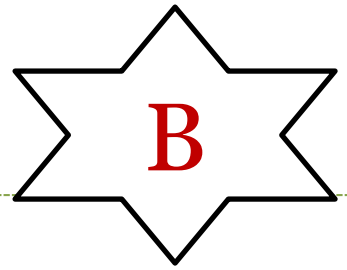
x = (15% 2 > 0) ? 0 : 1;

The value of x is

a) 7.5 b) 7

c) 0 d) 1

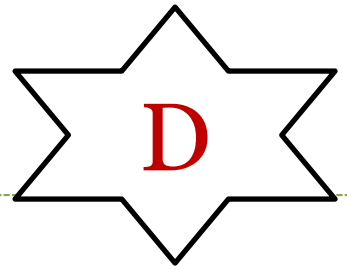
MCQ




```
for (int line = 1; line <  4; line++)
{
   for (int n = line; n >= 1; n--)
          cout << n;
          cout << endl;
}
```

a)	1 12 123	b)	1 21 321
c)	1 1 1	d)	1 2 3

MCQ

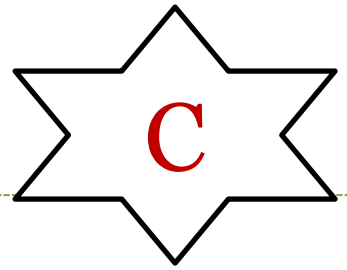



```
int num1, num2, x =1;
cin >> num1 >> num2;
while ((x*num1) % num2 != 0)
        x++;
cout << num1 * x;
```

This code calculates … of the two numbers

a) Highest common divisor
b) Highest common multiple
c) Least common divisor
d) Least common multiple

MCQ



Rules of operator precedence

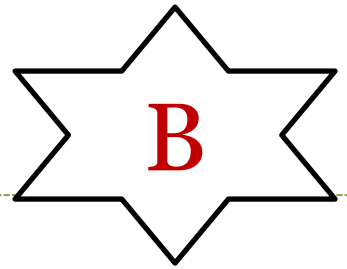
a) Addition, multiplication, parentheses

b) Multiplication, modulus, parentheses

c) Parentheses, division, subtraction

d) Parentheses, addition, division

MCQ



Preprocessor commands start with

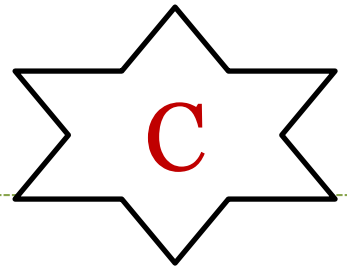
a) //

b) #

c) /*

d) DO

MCQ




```
int sum = 0, n;
cin >> n;
for (int i = 0; i <= n; i++)
        sum += pow (i, 2);
```

This statement calculates the series:

a) $1 + 2^2 + 2^3 + 2^4 + \ldots + 2^n$

b) $2^2 + 2^3 + 2^4 + \ldots + 2^n$

c) $1 + 2^2 + 3^2 + 4^2 + \ldots + n^2$

d) $2^2 + 3^2 + 4^2 + \ldots + n^2$

Correct and Justify

```
int x = -7;
while (x <= -10)
        cout << ++x;
```

Correction:

1. Use >= instead of <=
2. Use - -x instead of ++x

Correct and Justify

Calculate factorial for entered number

For example: 3! = 6 and 4! = 24.

```
int factorial, number;
cin >> number;
for (int i = 0; i <= number; i++)
{
    factorial += number;
    cout << factorial;
}
```

Correction

1. Initialize factorial to 1
2. Initialize i to 1
3. Use *= i instead of += number
4. Move last statement outside the loop

Correct and Justify

```
number,  x,  y;
(number  =  9) ?  x++  :  y-- ;
cout << x << y << endl;
```

Solution:

1. Declare variables
2. Initialize variables
3. Use == instead of =

Correct and Justify

```
int num = 12.5;
do {
    cout << "Enter a number: ";
    cin   >> num;
    }  while ( num >= 10 )
```

Solution:
- Use float instead of int
- Add ; at end of the statement

True/False and Correct

```
int rank;
cin >> rank;
switch (rank)
{
case 1:
        cout << "Gold\n";
        break;
case 2:
        cout << "Silver\n";
        break;
case 3:
        cout << "bronze\n";
        break;
case 4:
case 5:
        cout << "Certificate\n";
        break;
default:
        cout << "See you next competition\n";
}
```

True/False and Correct

1.The compiler displays a syntax error that no statements are executed in case 4.

2.The default statement is optional at the end of the switch statement.

3.The break statement is mandatory at the end of each case.

4.When the user enters 5, the program displays: Certificate.

5.When the user enters 2, the program displays: Silver, then on the following line it displays: See you next competition.

True/False: Solution

1. False.

   It falls to the next case to execute body of case 5.

2. True.

3. False.

   Break is optional at end of each case.

4. True.

5. False.

   It displays Silver only.

Correct Following False Statements

1. Linker substitutes text which includes other files or define constants.
2. Writing variables to memory is nondestructive.
3. The break statement causes the program to skip the rest of the loop in the current iteration and jumps to next iteration.

Correction

1. Pre-processor substitutes text which includes other files or define constants.

2. Reading variables from memory is nondestructive.

3. The continue statement causes the program to skip the rest of the loop in the current iteration and jumps to next iteration.

Display Output

```
int counter = 55;
for (int i = 3;  i <= 5;  i++)
      for ( int j = 12;  j <= 14;  j++)
           cout << i << " and "<< j<< endl;
          ++counter;
cout << counter;
```

56

Output:

3 and 12
3 and 13
3 and 14

4 and 12
4 and 13
4 and 14

5 and 12
5 and 13
5 and 14

Display Output

```
int counter = 55;
for (int i = 3;  i <= 5;  i++)
{
      for ( int j = 12;  j <= 14;  j++)
          cout << i << " and "<< j<< endl;
          ++counter;
}
cout << counter;
```

58

Output:

3 and 12
3 and 13
3 and 14

4 and 12
4 and 13
4 and 14

5 and 12
5 and 13
5 and 14

Display Output

```
int counter = 55;
for (int i = 3;  i <= 5;  i++)
{
     for ( int j = 12;  j <= 14;  j++) {
         cout << i << " and "<< j<< endl;
        ++counter;
      }
}
cout << counter;
```

64

Output:

3 and 12
3 and 13
3 and 14

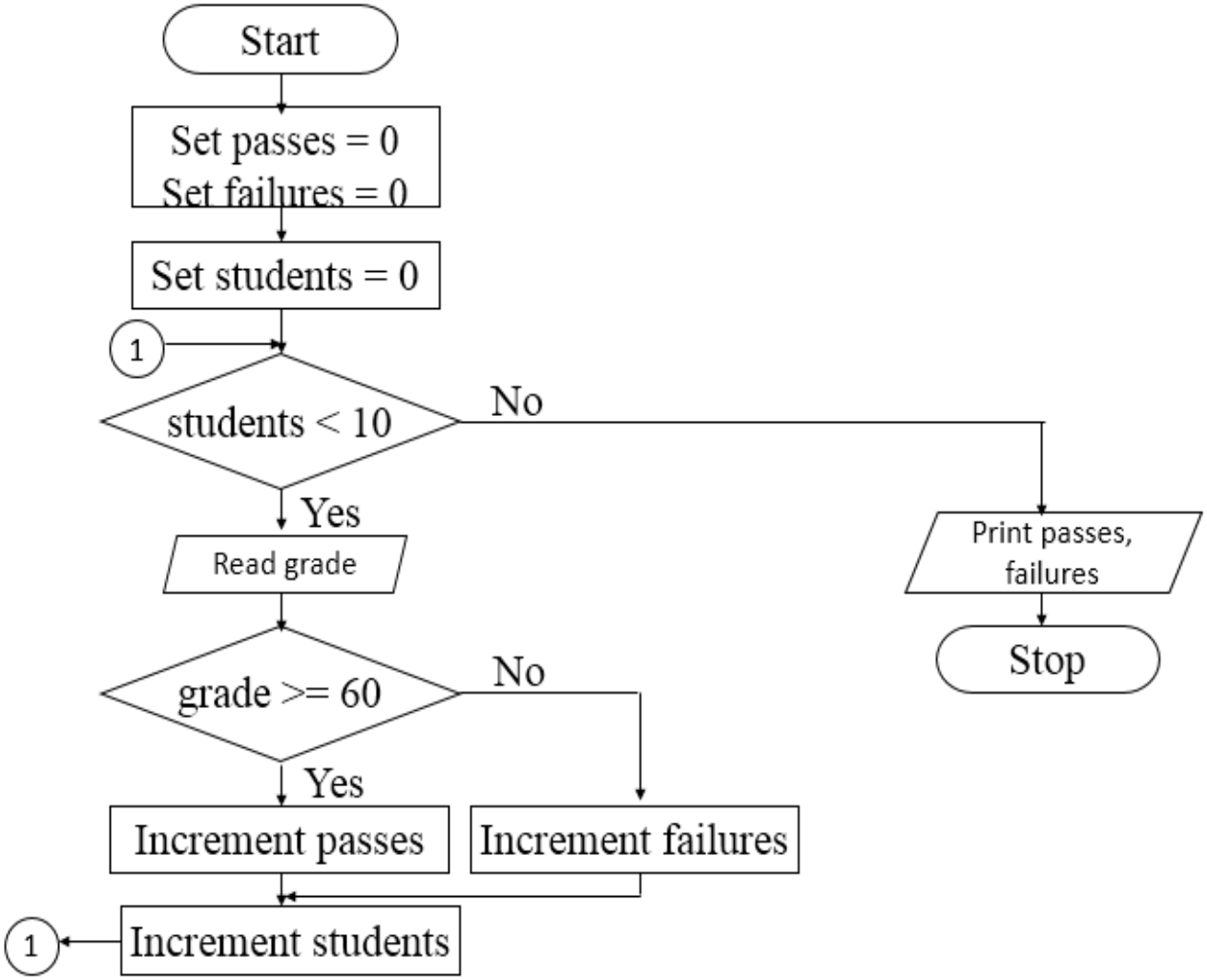
4 and 12
4 and 13
4 and 14

5 and 12
5 and 13
5 and 14

Draw a Flowchart

Design an algorithm that reads the exam results for 10 students and displays the numbers of passed and failed students. A student passes if his mark is greater than or equal to 60.

Flowchart


Start
Set passes = 0
Set failures = 0
Set students = 0
1
students < 10
No
Yes
Read grade
Print passes, failures
Stop
grade >= 60
No
Yes
Increment passes
Increment failures
1
Increment students

Write C++ Program

- Write a C++ program that calculates fines for speedy vehicles on the highway. It reads N persons' vehicle types and speeds in order to calculate the fines according to the following table.

Type	Speed	Fine
Car	121-140	500 LE
	141-180	1000 LE
	181-240	1500 LE
Bus	81-120	1000 LE
	121-160	2000 LE

Speedy Vehicles

```
float speed;
char vehicleType;
int N;

cout << "Enter the number of violations:\n";
cin  >> N;

for (int i = 0; i < N; i++)
{
cout << "Enter vehicle type:\n";
cin  >> vehicleType;
cout << "Enter speed:\n";
cin  >> speed;
```

```
if (vehicleType == 'c' || vehicleType == 'C')
        if (speed >= 121 && speed <= 140)
                cout << "Fine: 500 " << " for person number: "
                << i+1 << endl;
        else if (speed >= 141 && speed <= 180)
                cout << "Fine: 1000" << " for person number: "
                << i+1 << endl;
        else if (speed >= 181 && speed <= 240)
                cout << "Fine: 1500" << " for person number: "
                << i+1 << endl;
        else
                cout << "Undefined speed\n";
else if (vehicleType == 'b' || vehicleType == 'B')
        if (speed >= 81 && speed <= 120)
                cout << "Fine: 1000" << " for person number: "
                << i+1 << endl;
        else if (speed >= 121 && speed <= 160)
                cout << "Fine: 2000" << " for person number: "
                << i+1 << endl;
        else
                cout << "Undefined speed\n";
else
        cout << "Vechicle type is unknown\n";  } // end for
```

Another Solution

```
switch (vehicleType)
{
case 'c':
case 'C':
        if (speed >= 121 && speed <= 140)
                cout << "Fine: 500 " << " for person number: " << i+1 << endl;
        else if (speed >= 141 && speed <= 180)
                cout << "Fine: 1000" << " for person number: " << i+1 << endl;
        else if (speed >= 181 && speed <= 240)
                cout << "Fine: 1500" << " for person number: " << i+1 << endl;
        else
                cout << "Undefined speed\n";
        break;
case 'b':
case 'B':
        if (speed >= 81 && speed <= 120)
                cout << "Fine: 1000" << " for person number: " << i+1 << endl;
        else if (speed >= 121 && speed <= 160)
                cout << "Fine: 2000" << " for person number: " << i+1 << endl;
        else
                cout << "Undefined speed\n";
        break;
default:
        cout << "Vechicle type is unknown\n";
}
} // end for
```

Notes

- Mid-term grades will be available on my office door
- Date of mid-term retake will be announced soon
- Practical exam consists of one C++ program
- You **MUST** attend in the start of the time slot of your section according to the **NEW** practical exam schedule

General Group Practical Exam Schedule

	Section	Exam Starts at
Saturday 23-12-2017	Sec (1)	8:00 am
	Sec (2)	9:30 am
	Sec (3)	11:00 am
	Sec (4)	12:00 pm
Sunday 24-12-2017	Sec (5)	8:00 am
	Sec (6)	9:30 am
	Sec (7)	11:00 am
	Sec (8)	12:00 pm
Monday 25-12-2017	Sec (9)	8:00 am
	Sec (10)	9:30 am
	Sec (11)	11:00 am
	Sec (12)	12:00 pm
	Sec (13)	1:00 pm
Tuesday 26-12-2017	Sec (14)	8:00 am
	Sec (15)	9:30 am
	Sec (16)	11:00 am
	Sec (17)	12:00 pm
	Sec (18)	1:00 pm
Wednesday 27-12-2017	Sec (19)	8:00 am
	Sec (20)	9:30 am
	Sec (21)	11:00 am
	Sec (22)	12:00 pm

Bioinformatics Group
Practical Exam Schedule

Sunday 24/12/2017

Time Slot	9 - 10	10 - 11	11- 12	12 - 1	1 - 2
Section Number (#id in attendance sheet)	Section 1 (#1-#20)	Section 1 (#21 - #29) Section 2 (#1 - #10)	Section 2 (#11 - #28)	Section 3 (#1 - #20)	Section 3 (#21 – #28) Section 4 (#1 - #10)

Monday 25/12/2017

Time Slot	9 - 10	10 - 11	11- 12
Section Number (#id in attendance sheet)	Section 4 (#11-#29)	Section 5 (#1 - #20)	Section 5 (#21 - #29)

Software Engineering Group
Practical Exam Schedule

Wednesday 27/12/2017

Time Slot	9 - 10	10 - 11	11- 12	12 - 1	1 - 2
Section Number (#id in attendance sheet)	Section 1 (#1-#20)	Section 1 (#21 - #32) Section 2 (#1 - #10)	Section 2 (#11 - #27)	Section 3 (#1 - #15)	Section 3 (#16 – #31)


Feedback

Course Content:

- Appropriate for the stated level of the class?
- Was effectively organized in a way that helped me learn?
- The lab complemented my understanding of the lectures?
- The lab provided the opportunity to practice the skills required in the course?

- Please identify area(s) where you think the course could be improved.

Instructor:

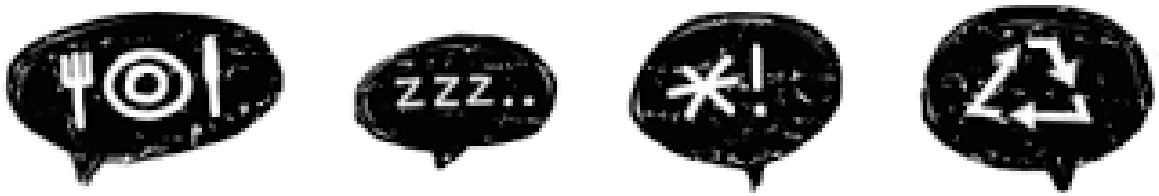
- Helped me define the goals and scope of the course?
- Presented content in an organized manner?
- Provided guidance for understanding course exercises?
- Encouraged student contributions and questions?
- How would you rate the overall effectiveness of the instructor's teaching?

TAs:

- Did TA explain lab material well?
- Is lab time enough for practice?

Student Self-Evaluation:

- How many lectures/labs did you attend?
- On average, how many hours per week have you spent on this course including attending classes, practicing, reviewing notes and any other course-related work?
- How satisfied were you with your effort in this course?



eat. sleep. revise. And repeat.