Kafrelsheikh University		**Date: 12 /1/2017**
Faculty of Engineering		**Time allowed: 3 hours**
Electrical Engineering Dept.		**Full Mark: 90**
Year: 3rd Elec. (Computers & Systems)	**Subject: Compilers**	**Final Term Exam**

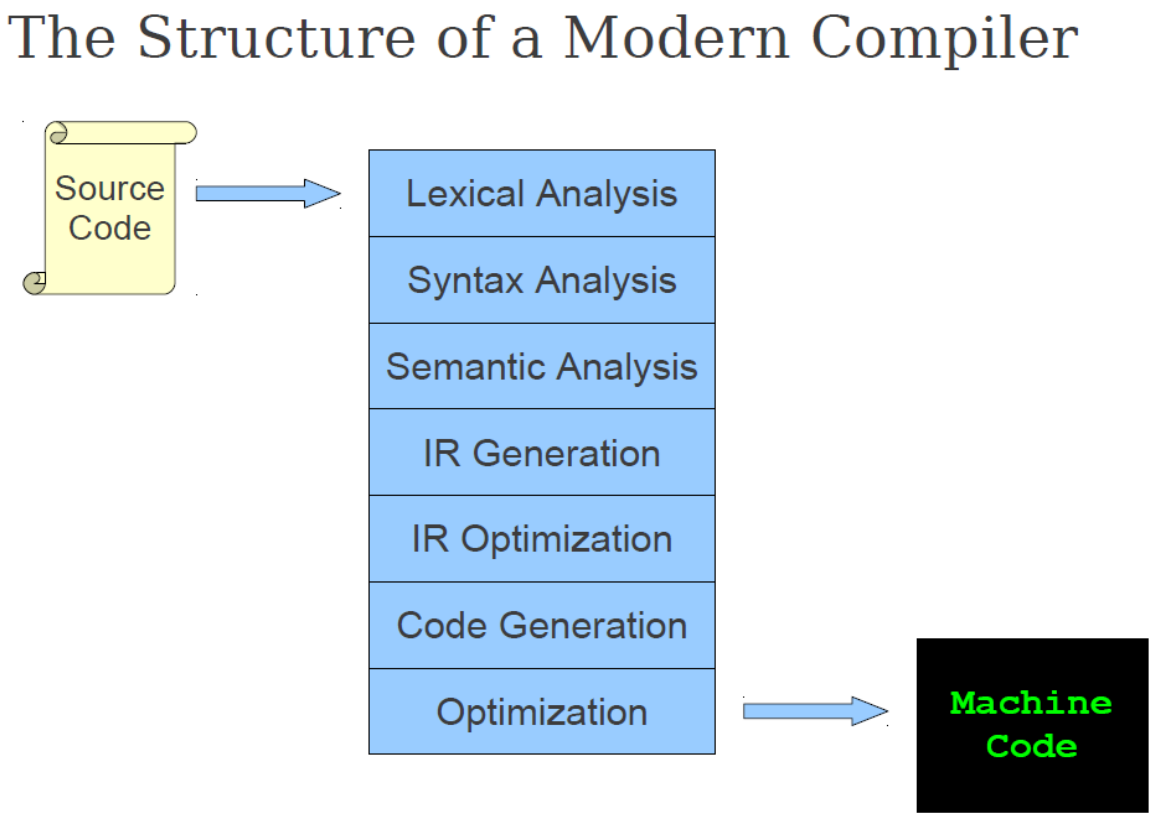
Answer as much as you can:

Question [1]: (35 marks)

a) **What is a compiler?**

Compiler is a program that *translates* a program in one language to another language: [Compilers: Translate a source (human-writable) program to an executable (machine-readable) program]

State the compiler phases (Explain by diagram.)



b) **What are the different kinds of errors encountered during compilation?**

1) **Lexical errors**
2) **Syntax errors**
3) **Semantic errors**
4) **Logical errors**

c) **What is the major stages in compilation?**

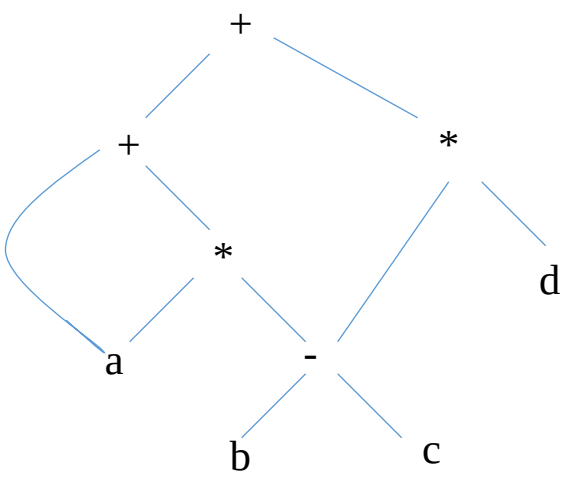
1) **Lexical analysis,**
2) **Syntax analysis,**
3) **Semantic analysis,**
4) **Code generation**

d) Define the following: Basic blocks and flow graphs.

- **Partition the intermediate code into basic blocks**
  - **The flow of control can only enter the basic block through the first instruction in the block. That is, there are no jumps into the middle of the block.**
  - **Control will leave the block without halting or branching, except possibly at the last instruction in the block.**
- **The basic blocks become the nodes of a flow graph**

e) What is happen when applying the three address code on the following expression:
Exp = (a+a*(b-c))+d*(b-c)

In a three address code there is at most one operator at the right side of an instruction



t1 = b – c
t2 = a * t1
t3 = a + t2
t4 = t1 * d
t5 = t3 + t4

Exp = (a+a*(b-c))+d*(b-c)

<u>Question [2]: (35 marks)</u>

a) Compile the following code (State step by step according to the compiler phases):

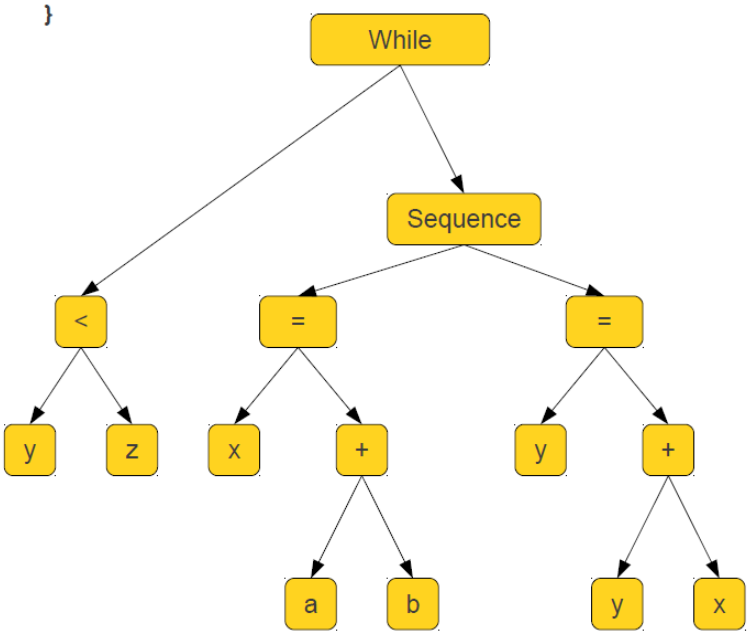
```
while (y < z) {
    int x = a + b;
    y += x;
}
```

```
while (y < z) {
    int x = a + b;
    y += x;
}
```

```
T_While
T_LeftParen
T_Identifier y
T_Less
T_Identifier z
T_RightParen
T_OpenBrace
T_Int
T_Identifier x
T_Assign
T_Identifier a
T_Plus
T_Identifier b
T_Semicolon
T_Identifier y
T_PlusAssign
T_Identifier x
T_Semicolon
T_CloseBrace
```

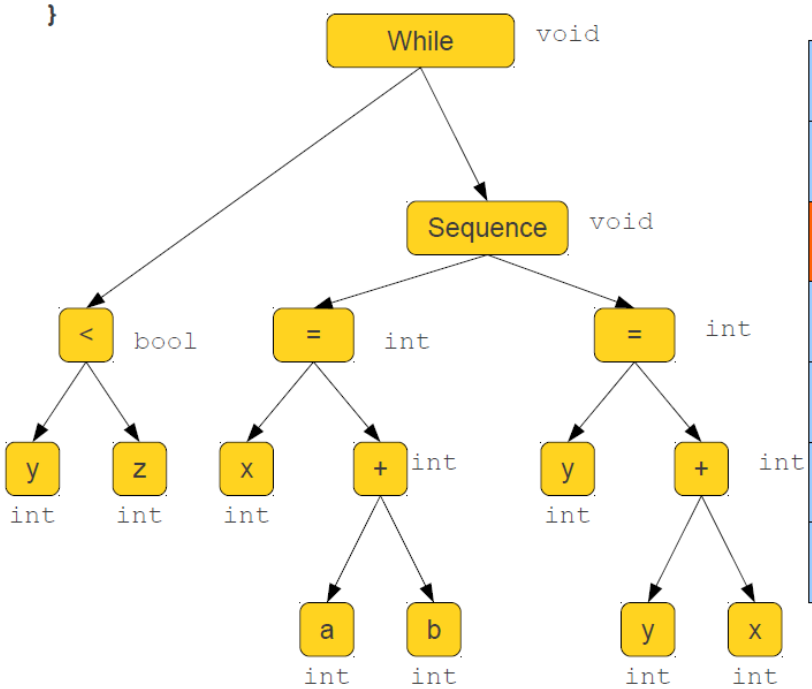
- **Lexical Analysis**
- Syntax Analysis
- Semantic Analysis
- IR Generation
- IR Optimization
- Code Generation
- Optimization

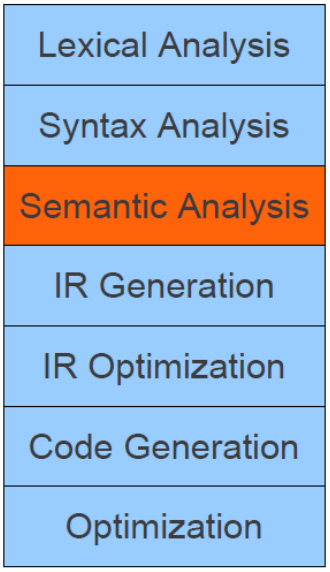
```
while (y < z) {
    int x = a + b;
    y += x;
}
```



- Lexical Analysis
- **Syntax Analysis**
- Semantic Analysis
- IR Generation
- IR Optimization
- Code Generation
- Optimization

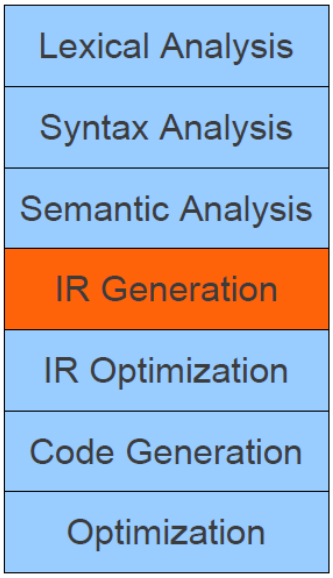
```
while (y < z) {
    int x = a + b;
    y += x;
}
```





```
while (y < z) {
    int x = a + b;
    y += x;
}
```

```
Loop: x    = a + b
      y    = x + y
      _t1  = y < z
      if _t1 goto Loop
```



```
while (y < z) {
    int x = a + b;
    y += x;
}
```

```
       x    = a + b
Loop:  y    = x + y
       _t1  = y < z
       if _t1 goto Loop
```

Lexical Analysis
Syntax Analysis
Semantic Analysis
IR Generation
IR Optimization
Code Generation
Optimization

```
while (y < z) {
    int x = a + b;
    y += x;
}
```

```
       add $1, $2, $3
Loop:  add $4, $1, $4
       slt $6, $4, $5
       beq $6, loop
```

Lexical Analysis
Syntax Analysis
Semantic Analysis
IR Generation
IR Optimization
Code Generation
Optimization

```
while (y < z) {
    int x = a + b;
    y += x;
}
```

```
       add $1, $2, $3
Loop:  add $4, $1, $4
       blt $4, $5, loop
```

b) State the forms of three address instructions.

<u>The forms of three address instructions are:</u>

x = y op z
x = op y
x = y
goto L
if x goto L and ifFalse x goto L
if x relop y goto L
Procedure calls using:
param x
call p,n
y = call p,n
x = y[i] and x[i] = y
x = &y and x = *y and *x =y

c) State some examples of reported errors.

<u>Some examples of reported errors:</u>

- **Undeclared identifier**
- **Multiply declared identifier**
- **Wrong number or types of call**
- **Incompatible types for operation**
- **Break statement outside switch/loop**
- **Goto with no label**

d) What are the typical semantic errors?

<u>The typical semantic errors are:</u>

Multiple declarations: a variable should be declared (in the same scope) at most once
Undeclared variable: a variable should not be used before being declared
Type mismatch: type of the LHS of an assignment should match the type of the RHS
Wrong arguments: methods should be called with the right number and types of arguments

e) **Why to separate Lexical analysis and parsing?**
- **Simplicity of design**
- **Improving compiler efficiency**
- **Enhancing compiler portability**

Question [3]: (20 marks)

What is the output of the following C programs?

a)

```
#include<stdio.h>
#include<stdlib.h>
void main()
{ int x, i,y=0;
for (i=0; i<5; i++)
{   x=rand()%10;
    printf("==> %d\n", x); }
}
```

The output is:

==>1
==>7
==> 4
==> 0
==> 9

Any number belongs to [0,9]

---

b)

```
#include <stdio.h>
void f1();
void f2();
void g();
void main()
{ int x = 0; f1(); g(); f2(); }
void f1() { int x = 10; g(); }
void f2() { int x = 20; f1(); g(); }
void g() { printf("x"); }
```

The output is:
xxxx