

Program Comprehension for Domain-Specific Programming Languages

QUESTIONNAIRE 3

GraphViz

Domain-Specific Programming Language

Name:	
Class:	
University:	
Date:	

Start time:	
--------------------	--

Question 1

Marks: 0 / 1

QL011 GraphViz-DSPL-CompilerConstruction: Please select correct GraphViz statements (without syntax errors):

Choose one answer.

- a. graph Compiler 1. {
 Source\nCode -> Compiler;
 Compiler -> Object\nCode;
 }
- b. digraph "Compiler 1." {
 graph rankdir LR;
 "Source\nCode" = "Compiler";
 "Compiler" = "Object\nCode";
 }
- c. digraph "Compiler 1." {
 graph [rankdir=LR];
 "Source\nCode" -> "Compiler";
 "Compiler" -> "Object\nCode";
 }
- d. graph:
 Source\nCode=> Compiler
 Compiler=> Object\nCode
- e. digraph "Compiler 1." {
 graph [rankdir=LR];
 Source\nCode -> Object\nCode [label= Compiler];
 }

Question 2

Marks: 0 / 1

QL012 GraphViz-DSPL-BranchingGame: Please select correct GraphViz statements (without syntax errors):

Choose one answer.

- a. graph "Branch 1." {
 start [shape = point peripheries = 2]
 end [shape = point peripheries = 2]
 "1" [shape = box width = 0.5 height = 0.5]
 "+3" [shape = box width = 0.5 height = 0.5]

 start -> "1" [arrowhead = open arrowtail = none]
 "1" -> "+3" [arrowhead = open arrowtail = none]
 "+3" -> end [arrowhead = open arrowtail = none]
 }
- b. digraph "Branch 1." {
 graph [rankdir=TD];
 start [shape = point, peripheries = 2];
 end [shape = point, peripheries = 2];
 "1" [shape = box , width = 0.5, height = 0.5];
 "+3" [shape = box , width = 0.5, height = 0.5];

 start -> "1" [arrowhead = open , arrowtail = none];
 "1" -> "+3" [arrowhead = open , arrowtail = none];
 }

```
"+3" -> end [arrowhead = open , arrowtail = none];
}
```

```
c. digraph "Branch 1." {
  graph [rankdir=TD];
  start      : shape      = point peripheries = 2      ;
  end        : shape      = point peripheries = 2      ;
  "1"        : shape      = box   width = 0.5 height = 0.5;
  "+3"       : shape      = box   width = 0.5 height = 0.5;

  start -> "1" : arrowhead = open arrowtail = none;
  "1"   -> "+3" : arrowhead = open arrowtail = none;
  "+3"  -> end  : arrowhead = open arrowtail = none;
}
```

```
d. digraph "Branch 1." {
  graph [rankdir=TD];
  start      [shape      = point peripheries = 2      ];
  end        [shape      = point peripheries = 2      ];
  '1'        [shape      = box   width = 0.5 height = 0.5];
  '+3'       [shape      = box   width = 0.5 height = 0.5];

  start => '1' [arrowhead = open arrowtail = none];
  '1'   => '+3' [arrowhead = open arrowtail = none];
  '+3'  => end [arrowhead = open arrowtail = none];
}
```

```
e. digraph "Branch 1." {
  graph : rankdir=TD
  start      : shape      = point, peripheries = 2      ]
  end        : shape      = point, peripheries = 2      ]
  '1'        : shape      = box  , width = 0.5, height = 0.5]
  '+3'       : shape      = box  , width = 0.5, height = 0.5]

  start -> '1' : arrowhead = open , arrowtail = none]
  '1'   -> '+3' : arrowhead = open , arrowtail = none]
  '+3'  -> end  : arrowhead = open , arrowtail = none]
}
```

Question 3

Marks: 0 / 1

QL021 GraphViz-DSPL-CompilerConstruction: Please select GraphViz program with no sense (unreasonable – incorrect compiler diagram):

Choose one answer.

- a. digraph "Compiler 2." {
graph [rankdir=LR];
node [shape=plaintext];
0 -> 1 [label="Object\nCode"];
1 -> 2 [label="Syntax\nAnalysis"];
2 -> 3 [label="Symbol\nTable\n"];
}
- b. digraph "Compiler 2." {
graph [rankdir=LR];
node [shape=plaintext];
"Lexical\nAnalysis" -> "Syntax\nAnalysis";
"Syntax\nAnalysis" -> "Semantic\nAnalysis";
}
- c. digraph "Compiler 2." {
graph [rankdir=LR];
subgraph cluster_0 {
graph [label="Compiler" style=dashed];
"Front End" [shape=box];
"Back End" [shape=box];
}
"Source\nCode" -> "Front End";
"Front End" -> "Back End";
"Back End" -> "Object\nCode";
}
- d. digraph "Compiler 2." {
graph [rankdir=LR];
0 [label = "Source Code", shape=plaintext];
1 [label = "Lexical Analysis"];
2 [label = "Symbol Table", shape=plaintext];
0 -> 1;
1 -> 2;
}
- e. digraph "Compiler 2." {
graph [rankdir = LR];
node [shape = plaintext];
0 -> 1 [label = "Intermediate\nCode"];
1 -> 2 [label = "Optimization"];
2 -> 3 [label = "Optimized\nCode"];
}

Question 4

Marks: 0 / 1

QL022 GraphViz-DSPL-BranchingGame: Please select GraphViz program with no sense (unreasonable – incorrect branching diagram):

Choose one answer.

- a. digraph "Branch 2." {
graph [rankdir=TD];
{
ordering = "out";
"start" [shape = box];
"end" [shape = box];
"1" [shape = box , width = 0.5, height = 0.5];
"+2" [shape = box , width = 0.5, height = 0.5];
"+3" [shape = box , width = 0.5, height = 0.5];

"start" -> "1" [arrowhead = open , arrowtail = none];
"1" -> "+2" [arrowhead = open , arrowtail = none];
"1" -> "+3" [arrowhead = open , arrowtail = none];
"+2" -> "end" [arrowhead = open , arrowtail = none];
"+3" -> "end" [arrowhead = open , arrowtail = none];
};
{ rank = same; "+2"; "+3"; };
}
- b. digraph "Branch 2." {
graph [rankdir=TD];
{
ordering = "out";
start [shape = point, peripheries = 2];
end [shape = point, peripheries = 2];
"0" [shape = box , width = 0.5, height = 0.5];
"+3" [shape = box , width = 0.5, height = 0.5];
"+5" [shape = box , width = 0.5, height = 0.5];

start -> end [arrowhead = open , arrowtail = none];
"0" -> "+3" [arrowhead = open , arrowtail = none];
"0" -> "+5" [arrowhead = open , arrowtail = none];
"+3" -> end [arrowhead = open , arrowtail = none];
"+5" -> end [arrowhead = open , arrowtail = none];
};
{ rank = same; "+3"; "+5"; };
}
- c. digraph "Branch 2." {
graph [rankdir=TD];
{
ordering = "out";
start [shape = point, peripheries = 2];
end [shape = point, peripheries = 2];
"0" [shape = box , width = 0.5, height = 0.5];
"+2" [shape = box , width = 0.5, height = 0.5];
"+4" [shape = box , width = 0.5, height = 0.5];

start-> "0" [arrowhead = open , arrowtail = none];
"0" -> "+2" [arrowhead = open , arrowtail = none];
"+2" -> "+4" [arrowhead = open , arrowtail = none];
"0" -> end [arrowhead = open , arrowtail = none];
"+4" -> end [arrowhead = open , arrowtail = none];
};
{ rank = same; "+2"; "+4"; };
}
- d. digraph "Branch 2." {
graph [rankdir=TD];

```
{
  ordering = "out";
  start      [shape      = point, peripheries = 2      ];
  end        [shape      = point, peripheries = 2      ];
  "1"        [shape      = box   , width = 0.5, height = 0.5];
  "2"        [shape      = box   , width = 0.5, height = 0.5];
  "3"        [shape      = box   , width = 0.5, height = 0.5];

  start -> "1" [arrowhead = open , arrowtail = none];
  start -> "2" [arrowhead = open , arrowtail = none];
  start -> "3" [arrowhead = open , arrowtail = none];
  "1" -> end  [arrowhead = open , arrowtail = none];
  "2" -> end  [arrowhead = open , arrowtail = none];
  "3" -> end  [arrowhead = open , arrowtail = none];
};
{ rank = same; "1"; "2"; "3";      };
}

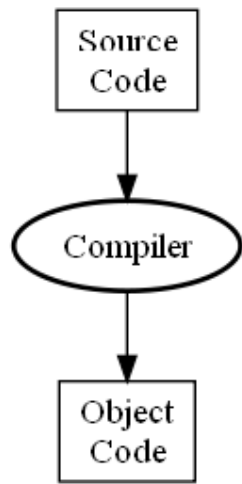
e. digraph "Branch 2." {
  graph [rankdir=TD];
  {
    ordering = "out";
    start      [shape      = point, peripheries = 2      ];
    end        [shape      = point, peripheries = 2      ];
    "0"        [shape      = box   , width = 0.5, height = 0.5];
    "+1"       [shape      = box   , width = 0.5, height = 0.5];
    "+2"       [shape      = box   , width = 0.5, height = 0.5];
    "+4"       [shape      = box   , width = 0.5, height = 0.5];

    start -> "0" [arrowhead = open , arrowtail = none];
    "0" -> "+1" [arrowhead = open , arrowtail = none];
    "+1" -> "+2" [arrowhead = open , arrowtail = none];
    "+1" -> "+4" [arrowhead = open , arrowtail = none];
    "+2" -> end  [arrowhead = open , arrowtail = none];
    "+4" -> end  [arrowhead = open , arrowtail = none];
  };
  { rank = same; "+2"; "+4";      };
}
}
```

Question 5

Marks: 0 / 1

QL031 GraphViz-DSPL-CompilerConstruction: Select program for the following figure:



Choose one answer.

- a.

```
digraph "Compiler 3." {
  graph [rankdir=TD];
  "Source\nCode" -> "Compiler";
  "Compiler" -> "Object\nCode";
  "Source\nCode" [shape=box3d]
  "Object\nCode" [shape=box3d]
  "Compiler" [style=filled]
}
```
- b.

```
digraph "Compiler 3." {
  node [shape=plaintext];
  0 [label="Compiler"];
  1 [label="Source\nCode"];
  0 -> 1;
  2 [label="Object\nCode"];
  0 -> 2;
}
```
- c.

```
digraph "Compiler 3." {
  graph [rankdir=LR];
  "Source\nCode";
  "Object\nCode";
  "Compiler" [style = bold];
  "Source\nCode" -> "Compiler";
  "Compiler" -> "Object\nCode";
}
```
- d.

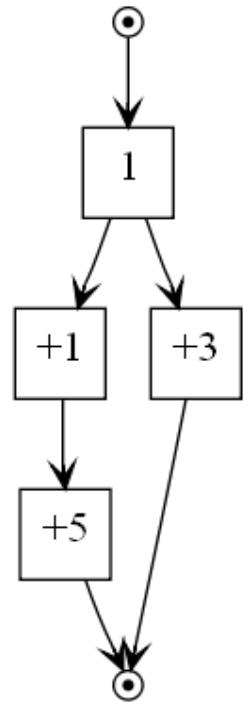
```
digraph "Compiler 3." {
  graph [rankdir=TD];
  "Source\nCode" [shape=box];
  "Object\nCode" [shape=box];
  "Compiler" [style=bold];
  "Source\nCode" -> "Compiler";
  "Compiler" -> "Object\nCode";
}
```
- e.

```
digraph "Compiler 3." {
  node [shape=plaintext];
  0 [label="Compiler", style=bold, shape=ellipse];
  1 [label="Source\nCode", shape=box];
  0 -> 1;
  2 [label="Object\nCode", shape=box];
  0 -> 2;
}
```

Question 6

Marks: 0 / 1

QL032 GraphViz-DSPL-BranchingGame: Select program for the following figure:



Choose one answer.

a. digraph "Branch 3." {
graph [rankdir=TD];
{
ordering = "out";
node [shape = box
start [shape = point, peripheries = 2];
end [shape = point, peripheries = 2];
"1" [width = 0.5 , height = 0.5];
"+1" [width = 0.5 , height = 0.5];
"+3" [width = 0.5 , height = 0.5];
"+5" [width = 0.5 , height = 0.5];
start-> "1" [arrowhead = open , arrowtail = none];
"1" -> "+1" [arrowhead = open , arrowtail = none];
"+1" -> "+3" [arrowhead = open , arrowtail = none];
"+3" -> "+5" [arrowhead = open , arrowtail = none];
"+5" -> end [arrowhead = open , arrowtail = none];
};
{ rank = same; "+1"; "+3"; "+5"; };
}

b. digraph "Branch 3." {
graph [rankdir=TD];
{
ordering = "out";
start ;
end ;
node1 [label = "1" , width = 0.5, height = 0.5];
node2 [label = "+1" , width = 0.5, height = 0.5];
node3 [label = "+3" , width = 0.5, height = 0.5];
node4 [label = "+5" , width = 0.5, height = 0.5];
start -> node1 [arrowhead = open , arrowtail = none];
node1 -> node2 [arrowhead = open , arrowtail = none];
node1 -> node3 [arrowhead = open , arrowtail = none];
node1 -> node4 [arrowhead = open , arrowtail = none];
}


```

node2 -> node4 [arrowhead = open , arrowtail = none];
node3 -> end   [arrowhead = open , arrowtail = none];
node4 -> end   [arrowhead = open , arrowtail = none];
};
{ rank = same; node2; node3;      };
}

c. digraph "Branch 3." {
graph [rankdir=TD];
{
ordering = "out";
start [          shape      = point, peripheries = 2          ];
end   [          shape      = point, peripheries = 2          ];
node1 [label = "1" , shape   = box   , width = 0.5, height = 0.5];
node2 [label = "+1", shape   = box   , width = 0.5, height = 0.5];
node3 [label = "+3", shape   = box   , width = 0.5, height = 0.5];
node4 [label = "+5", shape   = box   , width = 0.5, height = 0.5];

start -> node1 [arrowhead = open , arrowtail = none];
node1 -> node2 [arrowhead = open , arrowtail = none];
node1 -> node3 [arrowhead = open , arrowtail = none];
node3 -> node4 [arrowhead = open , arrowtail = none];
node2 -> end   [arrowhead = open , arrowtail = none];
node4 -> end   [arrowhead = open , arrowtail = none];
};
{ rank = same; node2; node4;      };
}

d. digraph "Branch 3." {
graph [rankdir=TD];
{
ordering = "out";
node [shape = box          ];
start ;
end   ;
"1" [width = 0.5, height = 0.5];
"+1" [width = 0.5, height = 0.5];
"+3" [width = 0.5, height = 0.5];
"+5" [width = 0.5, height = 0.5];

start -> "1" [arrowhead = open , arrowtail = none];
"1" -> "+1" [arrowhead = open , arrowtail = none];
"1" -> "+3" [arrowhead = open , arrowtail = none];
"+1" -> "+5" [arrowhead = open , arrowtail = none];
"+3" -> end [arrowhead = open , arrowtail = none];
"+5" -> end [arrowhead = open , arrowtail = none];
};
{ rank = same; "+3"; "+5";      };
}

e. digraph "Branch 3." {
graph [rankdir=TD];
{
ordering = "out";
start [shape      = point, peripheries = 2          ];
end   [shape      = point, peripheries = 2          ];
"1" [shape      = box   , width = 0.5, height = 0.5];
"+1" [shape      = box   , width = 0.5, height = 0.5];
"+3" [shape      = box   , width = 0.5, height = 0.5];
"+5" [shape      = box   , width = 0.5, height = 0.5];

start -> "1" [arrowhead = open , arrowtail = none];
"1" -> "+1" [arrowhead = open , arrowtail = none];
"1" -> "+3" [arrowhead = open , arrowtail = none];
}

```

```

"+1" -> "+5" [arrowhead = open , arrowtail = none];
"+3" -> end [arrowhead = open , arrowtail = none];
"+5" -> end [arrowhead = open , arrowtail = none];
};
{ rank = same; "+1"; "+3";      };
}

```

End time:

Question 7

Marks: 0 / 1

QC011 GraphViz-DSPL-BranchingGame: Please select valid figure of the following GraphViz program:

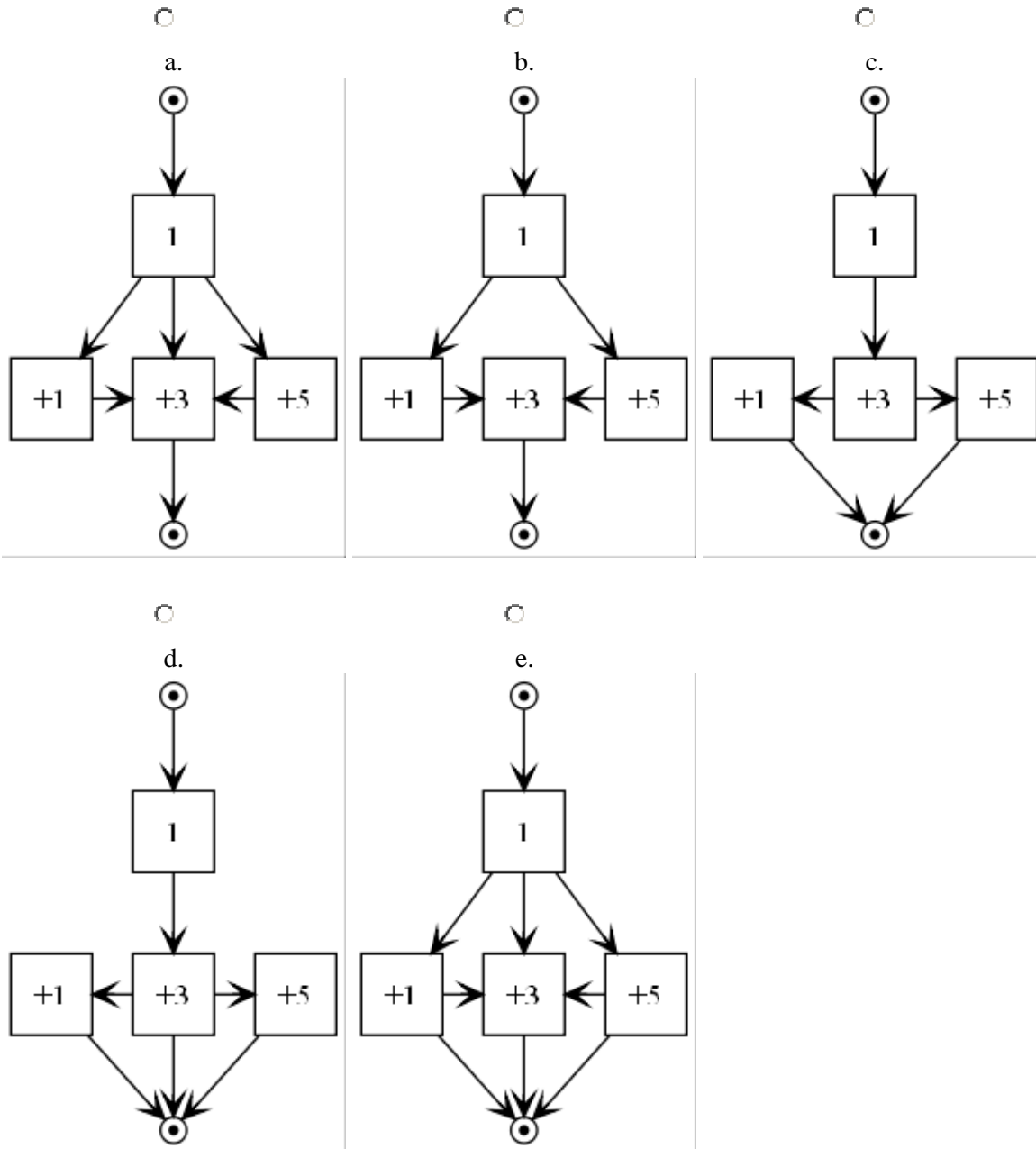
```

digraph "Branch 4." {
  graph [rankdir=TD];
  {
    ordering = "out";
    start      [shape      = point, peripheries = 2      ];
    end        [shape      = point, peripheries = 2      ];
    "1"        [shape      = box   , width = 0.5, height = 0.5];
    "+1"       [shape      = box   , width = 0.5, height = 0.5];
    "+3"       [shape      = box   , width = 0.5, height = 0.5];
    "+5"       [shape      = box   , width = 0.5, height = 0.5];

    start-> "1" [arrowhead = open , arrowtail = none];
    "1" -> "+3" [arrowhead = open , arrowtail = none];
    "+1" -> "+3" [arrowhead = none , arrowtail = open];
    "+3" -> "+5" [arrowhead = open , arrowtail = none];
    "+1" -> end [arrowhead = open , arrowtail = none];
    "+5" -> end [arrowhead = open , arrowtail = none];
  };
  { rank = same; "+1"; "+3"; "+5";      };
}

```

Choose one answer.



Question 8

Marks: 0 / 1

QC012 GraphViz-DSPL-CompilerConstruction: Please select valid figure of the following GraphViz

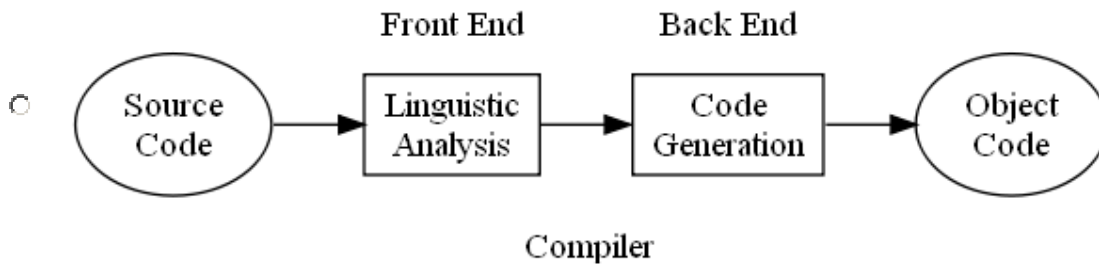
program:

```

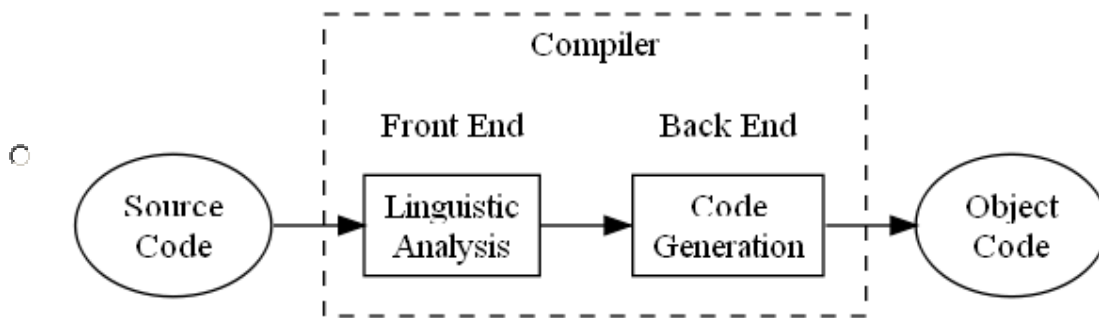
digraph "Figure 4." {
  graph [rankdir=LR];
  subgraph cluster_0 {
    graph [label=Compiler, style=dashed];
    subgraph cluster_0_1 {
      graph [color=white, label="Front End"];
      "Linguistic\nAnalysis" [shape=box];
    }
    subgraph cluster_0_2 {
      graph [color=white, label="Back End"];
      "Code\nGeneration" [shape=box];
    }
  }
  "Source\nCode" -> "Linguistic\nAnalysis";
  "Linguistic\nAnalysis" -> "Code\nGeneration";
  "Code\nGeneration" -> "Object\nCode";
}
    
```

Choose one answer.

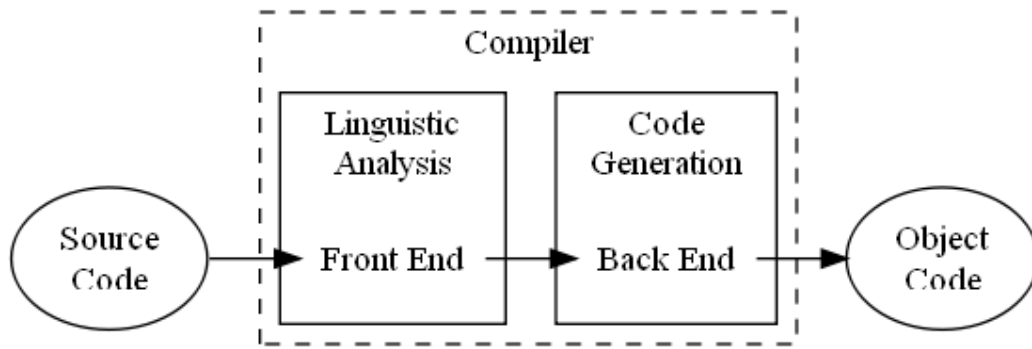
a.



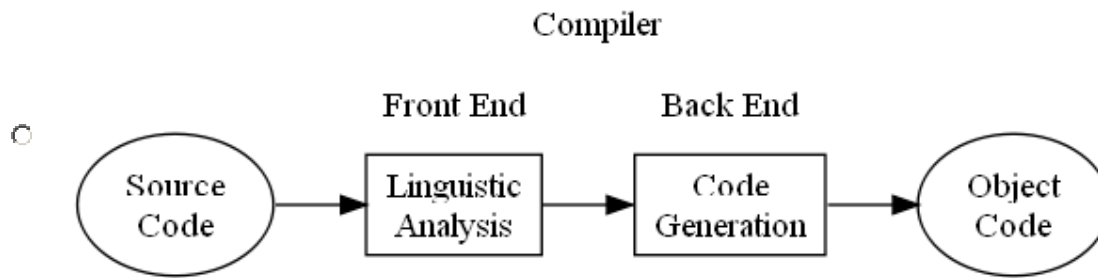
b.



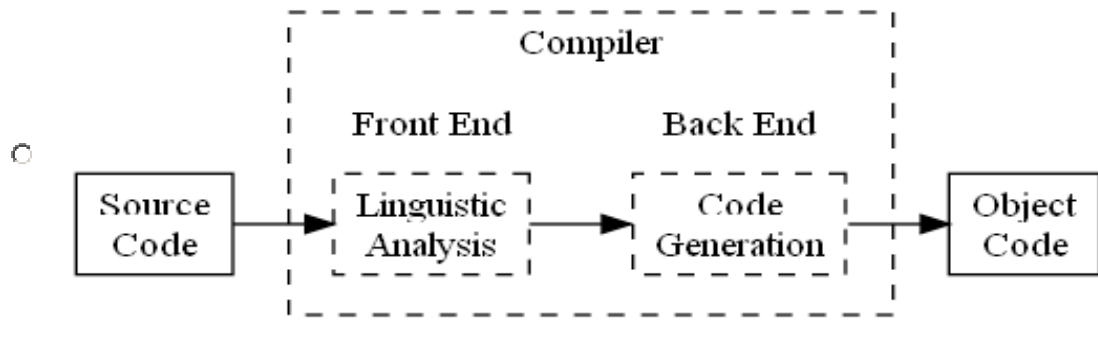
c.



d.



e.



Question 9

Marks: 0 / 1

QC021 GraphViz-DSPL-BranchingGame: Please select number of edge definitions within the following GraphViz program :

```
digraph "Branch 5." {
  graph [rankdir=TD];
  {
    ordering = "out";
    start      [shape = point, peripheries = 2      ];
    end        [shape = point, peripheries = 2      ];
    "1"        [shape = box  , width = 0.5, height = 0.5];
    "+1"       [shape = box  , width = 0.5, height = 0.5];
    "+3"       [shape = box  , width = 0.5, height = 0.5];
    "+4"       [shape = box  , width = 0.5, height = 0.5];
    "+5"       [shape = box  , width = 0.5, height = 0.5];
    "+2"       [shape = box  , width = 0.5, height = 0.5];

    start -> "1" [arrowhead = open , arrowtail = none];
    "1"   -> "+1" [arrowhead = open , arrowtail = none];
    "1"   -> "+3" [arrowhead = open , arrowtail = none];
    "+1"  -> "+5" [arrowhead = open , arrowtail = none];
    "+3"  -> "+4" [arrowhead = open , arrowtail = none];
    "+4"  -> "+2" [arrowhead = open , arrowtail = none];
    "+5"  -> "+2" [arrowhead = open , arrowtail = none];
    "+2"  -> end [arrowhead = open , arrowtail = none];
  };
  { rank = same; "+1"; "+3"; };
  { rank = same; "+4"; "+5"; };
}
```

Choose one answer.

- a. 10
- b. 4
- c. 0
- d. 8
- e. 1

Question 10

Marks: 0 / 1

QC022 GraphViz-DSPL-CompilerConstruction: Please select number of nodes with ellipse shape within the following GraphViz program :

```
digraph "Compiler 5." {
graph [ rankdir=TD label = "Compiler" ];
{
  "Linguistic\nAnalysis" -> "Code\nGeneration" ->
                        "Code\nOptimization" ->
                        "Object\nCode" ;
  "Linguistic\nAnalysis" [shape=box];
  "Object\nCode" [shape=box];
};
}
```

Choose one answer.

- a. 4
- b. 2
- c. 0
- d. 1
- e. 3

Question 11

Marks: 0 / 1

QC031 GraphViz-DSPL-BranchingGame: Select GraphViz program with the same result as program below:

```
digraph "Branch 6." {
graph [rankdir=TD];
{
  start [shape = point, peripheries = 2];
  end [shape = point, peripheries = 2];
  "1" [shape = box, width = 0.5, height = 0.5];
  "+1" [shape = box, width = 0.5, height = 0.5];
  "+3" [shape = box, width = 0.5, height = 0.5];
  "+4" [shape = box, width = 0.5, height = 0.5];
  "+5" [shape = box, width = 0.5, height = 0.5];

  start-> "1" [arrowhead = open, arrowtail = none];
  "1" -> "+1" [arrowhead = open, arrowtail = none];
  "+1" -> "+3" [arrowhead = open, arrowtail = none];
  "+3" -> "+4" [arrowhead = open, arrowtail = none];
  "+4" -> "+5" [arrowhead = open, arrowtail = none];
  "+5" -> end [arrowhead = open, arrowtail = none];
};
{ rank = same; "+1"; "+3"; };
{ rank = same; "+4"; "+5"; };
}
```

Choose one answer.

a. digraph "Branch 6." {
graph [rankdir=TD];
{
start [shape = point, peripheries = 2];
end [shape = point, peripheries = 2];
node [shape = box , width = 0.5, height = 0.5];

edge [arrowhead = open , arrowtail = none];
start-> "1" ;
"1" -> "+1" ;
"+1" -> "+3" ;
"+3" -> "+4" ;
"+4" -> "+5" ;
"+5" -> end ;
};
{ rank = same; "+1"; "+3"; };
{ rank = same; "+4"; "+5"; };
}

b. digraph "Branch 6." {
graph [rankdir=TD];
{
start [shape = point, peripheries = 2];
end [shape = point, peripheries = 2];

start-> "1" ;
"1" -> "+1" ;
"+1" -> "+3" ;
"+3" -> "+4" ;
"+4" -> "+5" ;
"+5" -> end ;
};
}

c. digraph "Branch 6." {
graph [rankdir=TD];
{
ordering = "out";
start [shape = point, peripheries = 2];
end [shape = point, peripheries = 2];

start-> "1" ;
"1" -> "+1" ;
"+1" -> "+3" ;
"+3" -> "+4" ;
"+4" -> "+5" ;
"+5" -> end ;
};
{ rank = same; "+1"; "+3"; };
{ rank = same; "+4"; "+5"; };
}

d. digraph "Branch 6." {
graph [rankdir=TD];
start [shape = point, peripheries = 2];
end [shape = point, peripheries = 2];
"1" [shape = box , width = 0.5, height = 0.5];
"+1" [shape = box , width = 0.5, height = 0.5];
"+3" [shape = box , width = 0.5, height = 0.5];
"+4" [shape = box , width = 0.5, height = 0.5];


```

"+5"      [shape      = box      , width = 0.5, height = 0.5];

start-> "1"  [arrowhead = open , arrowtail = none];
"1"  -> "+1" [arrowhead = open , arrowtail = none];
"+1" -> "+3" [arrowhead = open , arrowtail = none];
"+3" -> "+4" [arrowhead = open , arrowtail = none];
"+4" -> "+5" [arrowhead = open , arrowtail = none];
"+5" -> end  [arrowhead = open , arrowtail = none];
}

```

```

e. digraph "Branch 6." {
  graph [rankdir=TD];
  {
    start [shape      = point, peripheries = 2      ];
    end   [shape      = point, peripheries = 2      ];
    node  [shape      = box   , width = 0.5 , height = 0.5];

    node  [arrowhead = open , arrowtail = none];
    start -> "1" -> "+3" -> "+1" -> "+5" -> "+4" -> end;
  };
  { rank = same; "+3"; "+1"; };
  { rank = same; "+5"; "+4"; };
}

```

Question 12

Marks: 0 / 1

QC032 GraphViz-DSPL-CompilerConstruction: Select GraphViz program with the same result as program below:

```

digraph "Compiler 6." {
  graph [ rankdir=TD label = "Compiler"];
  {
    ordering = "out";
    node [shape=box];
    "Symbol\nTable"      -> "Linguistic\nAnalysis" [arrowhead=none];
    "Linguistic\nAnalysis" [shape=ellipse ];
    "Linguistic\nAnalysis" -> "Error\nHandler"      [arrowhead=none];
    "Source\nCode"        -> "Linguistic\nAnalysis";
    "Linguistic\nAnalysis" -> "Code\nGeneration" ->
      "Code\nOptimization"->
      "Object\nCode"      ;
    "Code\nGeneration"   [shape=ellipse ];
    "Code\nOptimization" [shape=ellipse ];
    "Code\nGeneration"   -> "Symbol\nTable"      [arrowhead=none];
    "Code\nGeneration"   -> "Error\nHandler"      [arrowhead=none];
    "Code\nOptimization" -> "Symbol\nTable"      [arrowhead=none];
    "Code\nOptimization" -> "Error\nHandler"      [arrowhead=none];
  };
  {rank = same; "Symbol\nTable"; "Linguistic\nAnalysis"; "Error\nHandler";};
}

```

Choose one answer.

```

a. digraph "Compiler 6." {
  graph [ label = "Compiler"];
  {
    ordering="out";
    sc [label = "Source\nCode"      ];
    la [label = "Linguistic\nAnalysis"];
    st [label = "Symbol\nTable"     ];
    eh [label = "Error\nHandler"    ];
    cg [label = "Code\nGeneration"  ];
    co [label = "Code\nOptimization"];
    oc [label = "Object\nCode"     ];
  }
}

```

```

    sc -> la;
    st -> la;
    la -> eh;
    la -> cg -> co-> oc;
    cg -> st;
    cg -> eh;
    co -> st;
    co -> eh;
};
{rank = same; st; oc; eh; };
}
b. digraph "Compiler 6." {
graph [ rankdir=TD, label="Compile" ];
{
    ordering = "out";
    node [shape=box];
    "Symbol\nTable"      -> "Linguistic\nAnalysis" [arrowhead=none];
    "Linguistic\nAnalysis" [shape=ellipse ];
    "Linguistic\nAnalysis" -> "Error\nHandler"      [arrowhead=none];
    "Source\nCode"      -> "Linguistic\nAnalysis";
    "Linguistic\nAnalysis" -> "Code\nGeneration" ->
                                "Code\nOptimization"->
                                "Object\nCode"          ;
    "Code\nGeneration" [shape=ellipse ];
    "Code\nOptimization" [shape=ellipse ];
    "Code\nGeneration" -> "Symbol\nTable"      [arrowhead=none];
    "Code\nGeneration" -> "Error\nHandler"      [arrowhead=none];
    "Code\nOptimization" -> "Symbol\nTable"      [arrowhead=none];
    "Code\nOptimization" -> "Error\nHandler"      [arrowhead=none];
};
{rank = same; "Symbol\nTable" ;
                "Code\nGeneration";
                "Error\nHandler" ;
};
}
c. digraph "Compiler 6." {
graph [ label = "Compiler" ];
{
    ordering="out";
    sc [label = "Source\nCode"      , shape=box    ];
    la [label = "Linguistic\nAnalysis", shape=ellipse];
    st [label = "Symbol\nTable"      , shape=box    ];
    eh [label = "Error\nHandler"      , shape=box    ];
    cg [label = "Code\nGeneration"    , shape=ellipse];
    co [label = "Code\nOptimization"  , shape=ellipse];
    oc [label = "Object\nCode"        , shape=box    ];
    sc -> la;
    st -> la [arrowhead=none];
    la -> eh [arrowhead=none];
    la -> cg -> co-> oc;
    cg -> st [arrowhead=none];
    cg -> eh [arrowhead=none];
    co -> st [arrowhead=none];
    co -> eh [arrowhead=none];
};
}
d. digraph "Compiler 6." {
graph [ rankdir=LR label = "Compiler" ];
{
    node [shape=box];
    "Symbol\nTable"      -> "Linguistic\nAnalysis" [arrowhead=none];
    "Linguistic\nAnalysis" [shape=ellipse ];
    "Linguistic\nAnalysis" -> "Error\nHandler"      [arrowhead=none];
    "Source\nCode"      -> "Linguistic\nAnalysis";
};
}

```

```

"Linguistic\nAnalysis" -> "Code\nGeneration" ->
                        "Code\nOptimization"->
                        "Object\nCode"      ;

"Code\nGeneration"      [shape=ellipse ];
"Code\nOptimization"    [shape=ellipse ];
"Code\nGeneration"      -> "Symbol\nTable"    [arrowhead=none];
"Code\nGeneration"      -> "Error\nHandler"  [arrowhead=none];
"Code\nOptimization"    -> "Symbol\nTable"    [arrowhead=none];
"Code\nOptimization"    -> "Error\nHandler"  [arrowhead=none];
};
{rank = same; "Symbol\nTable";
  "Linguistic\nAnalysis";
  "Error\nHandler";
};
}
e. digraph "Compiler 6." {
graph [ rankdir=TD label = "Compiler"];
{
  ordering = "out";
  node [shape = box];
  st [label = "Symbol\nTable"                ];
  la [label = "Linguistic\nAnalysis", shape=ellipse];
  eh [label = "Error\nHandler"                ];
  sc [label = "Source\nCode"                  ];
  cg [label = "Code\nGeneration"             , shape=ellipse];
  co [label = "Code\nOptimization"           , shape=ellipse];
  oc [label = "Object\nCode"                  ];
  st -> la [arrowhead=none];
  la -> eh [arrowhead=none];
  sc -> la;
  la -> cg -> co -> oc;
  cg -> st [arrowhead=none];
  cg -> eh [arrowhead=none];
  co -> st [arrowhead=none];
  co -> eh [arrowhead=none];
};
{rank = same; st; la; eh;};
}

```

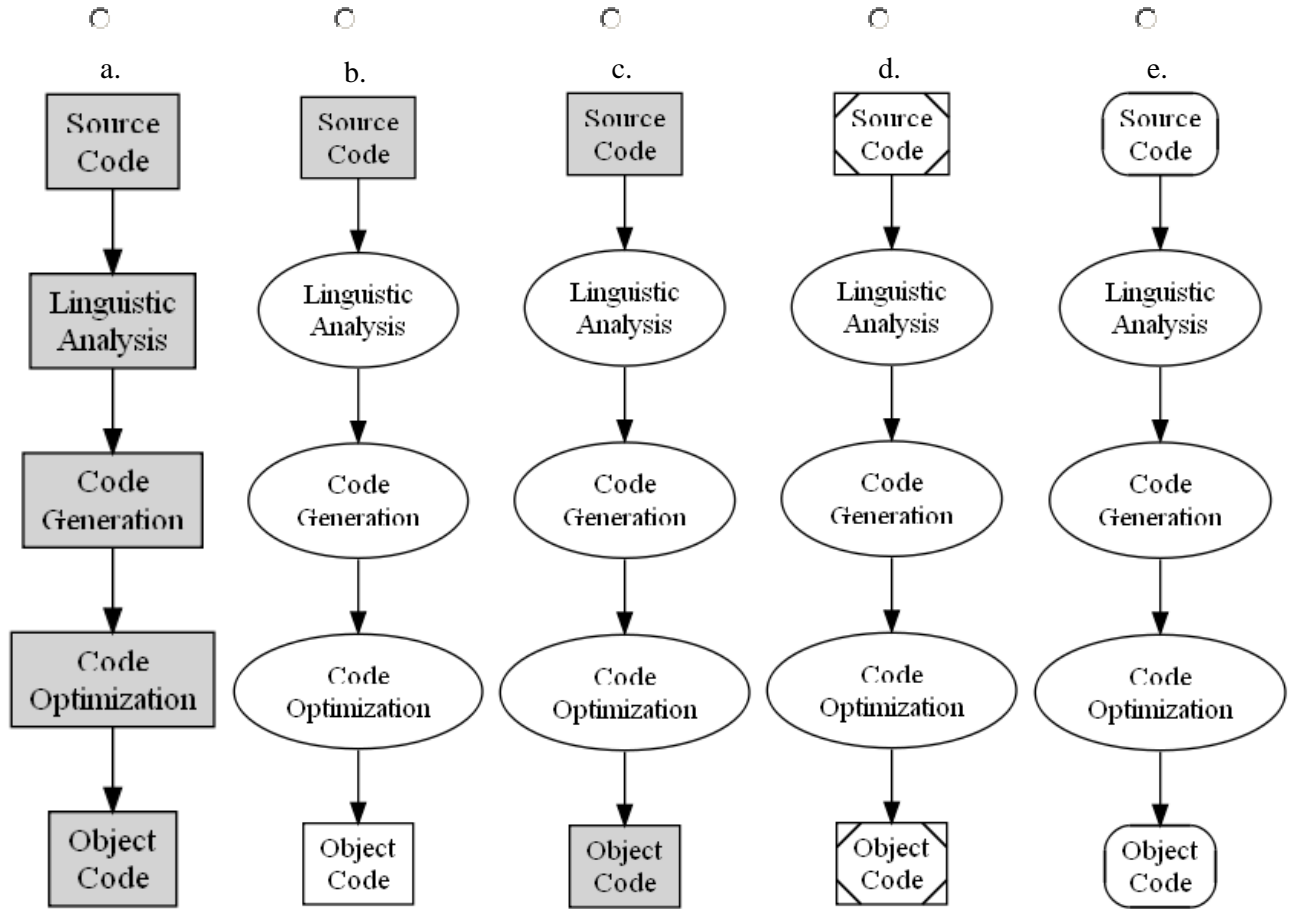
Question 13

Marks: 0 / 1

QC041 GraphViz-DSPL-CompilerConstruction: Please select valid figure of the following graphViz program. Notice, that the new construct *style* has been introduced to the program:

```
digraph "Compiler 7." {
"Source\nCode" [shape=box, style = filled];
"Source\nCode" -> "Linguistic\nAnalysis" -> "Code\nGeneration" ->
"Code\nOptimization"->"Object\nCode";
"Object\nCode" [shape=box, style = filled];
}
```

Choose one answer.



Question 14

Marks: 0 / 1

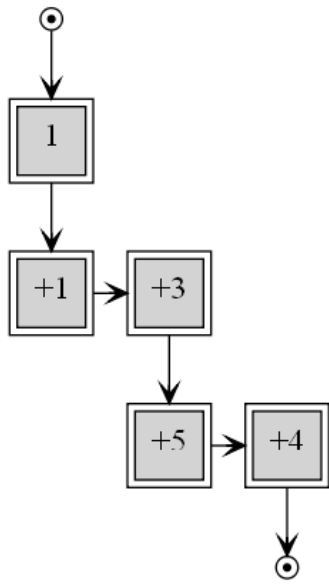
QC042 GraphViz-DSPL-BranchingGame: Please select valid figure of the following graphViz program.

Notice, that the new construct *polygon* and *sides* have been introduced to the program:

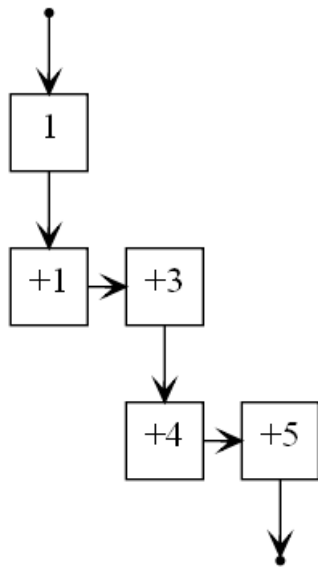
```
digraph "Branch 7." {
  graph [rankdir=TD];
  start [shape = point , peripheries = 2];
  end [shape = point , peripheries = 2];
  node [shape = polygon, sides = 4 , peripheries = 2 ,
        style = filled , width = 0.5, height = 0.5 ];
  edge [arrowhead=open , arrowtail=none, style=bold ];
  start -> "1";
  "1" -> "+1";
  "1" -> "+3";
  "+1" -> "+5";
  "+3" -> "+4";
  "+4" -> end ;
  "+5" -> end ;
}
```

Choose one answer.

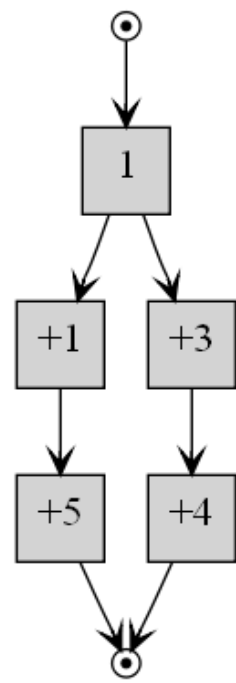
a.

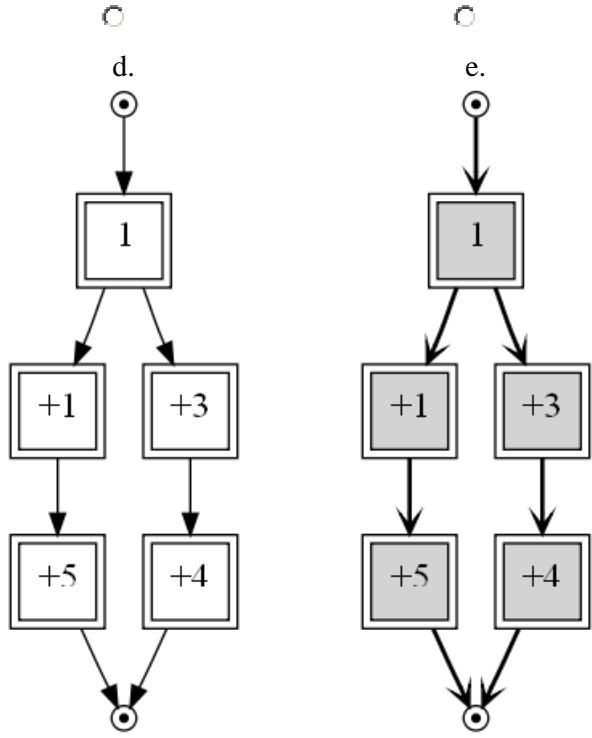


b.



c.





Question 15

Marks: 0 / 1

QC051 GraphViz-DSPL-BranchingGame: Please select number of nodes in following GraphViz program:

```
digraph "Branch 8." {
  graph [rankdir=TD]; /* Nodes direction is from top to down */
  {
    /* First node name is 'start', second node name 'end' */
    /* Both nodes are of type point and have two peripheries */
    start [shape = point, peripheries = 2 ];
    end [shape = point, peripheries = 2 ];
    /* Following nodes are of shape box, with specific size */
    node [shape = box , width = 0.5, height = 0.5];
    /* Following edges have arrow symbol in head side */
    edge [arrowhead = open , arrowtail = none];

    start -> "+3"; /* Edge between start and "+3" */
    start -> "+1"; /* Edge between start and "+1" */
    start -> "+5"; /* Edge between start and "+5" */
    "+3" -> "+2"; /* Edge between "+3" and "+2" */
    "+1" -> "+2"; /* Edge between "+1" and "+2" */
    "+5" -> "+2"; /* Edge between "+5" and "+2" */
    "+2" -> end ; /* Edge between "+2" and end */
  };

  /* Defined nodes will be in the same line */
  { rank = same; "+1"; "+3"; "+5"; };
}
```

Choose one answer.

- a. 6
- b. 4
- c. 7
- d. 3
- e. 8

Question 16

Marks: 0 / 1

QC052 GraphViz-DSPL-CompilerConstruction: Please select number of nodes in following GraphViz program:

```

digraph "Figure 8." {
  /* Nodes placement in graph from left ot right */
  graph [rankdir=LR];

  /* Graph Compiler has dashed box outline */
  subgraph cluster_0 {
    graph [label = Compiler, style=dashed ];

    /* Graph Compiler consists of subgraphs */
    subgraph cluster_0_1 {
      /* First subgraph Front end with box color white */
      graph [color = white , label="Front End" ];

      /* Record node is divided in two rows - first row is further
      /* divided in Lexical/Syntax/Semantic analysis
      linguisticAnalysis[shape = record,
        label = "{ <f0> Lexical\nAnalysis
                  |<f1> Syntax\nAnalysis
                  |<f2> Semantic\nAnalysis
                  }
                  |Liguistic Analysis"
      ];
    }

    /* Second subgraph Back end of Compiler graph */
    subgraph cluster_0_2 {
      /* Box color set to white */
      graph [color = white , label = "Back End" ];
      /* Node Code Generation inside subgraph Back end
      "Code Generation" [shape = box ];
    }
  }

  /* Edge between Source code and Lexical Analysis */
  "Source\nCode" -> linguisticAnalysis;

  /* Edge between Lexical Analysis and Code Generation */
  linguisticAnalysis-> "Code Generation" ;

  /* Edge between Code Generation and Object Code */
  "Code Generation" -> "Object\nCode" ;
}

```

Choose one answer.

- a. 7
- b. 6
- c. 5
- d. 4
- e. 3

End time:	
------------------	--

Question 17

Marks: 0 / 1

QE011 GraphViz-DSPL-BranchingGame: Expand arithmetic diagram with the connection between "1" and "+1" node. Also, expand diagram with the connection between "+1" and node "end".

```
digraph "Branch 9." {
  graph [rankdir=TD];
  {
    ordering = "out";
    start [shape = point, peripheries = 2 ];
    end [shape = point, peripheries = 2 ];
    node [shape = box , width = 0.5 , height = 0.5];

    edge [arrowhead = open , arrowtail = none];
    start-> "1" ;
    "1" -> "+3" ;
    "+1" -> "+3" ;
    "+3" -> "+5" ;
    "+3" -> "+2" ;
    "+2" -> end ;
    "+5" -> end ;
  };
  { rank = same; "+1"; "+3"; "+5"; };
}
```

Answer:

Question 18

Marks: 0 / 1

QE012 GraphViz-DSPL-CompilerConstruction: Expand compiler phases with the "Code Optimization" phase. This phase follows the "Code Generation" phase (inside "Compiler").

```
digraph "Figure 9." {
  graph [rankdir=LR];
  subgraph cluster_0 {
    graph [label = Compiler, style=dashed ];

    linguisticAnalysis [shape = record,
      label = "{ <f0> Lexical\nAnalysis
                |<f1> Syntax\nAnalysis
                |<f2> Semantic\nAnalysis
                }
                |Linguistic Analysis" ];
    "Code Generation" [shape = box ];
  }
  "Source\nCode"      -> linguisticAnalysis;
  linguisticAnalysis-> "Code Generation" ;
  "Code Generation" -> "Object\nCode"      ;
}
```

Answer:

Question 19

Marks: 0 / 1

QE021 GraphViz-DSPL-BranchingGame: Change the GraphViz program below in a way that node "+1" is no longer part of arithmetic diagram. Also remove all edges connected with node "+1".

```
digraph "Branch 10." {
  graph [rankdir=TD];
  {
    ordering = "out";
    start[shape = point, peripheries = 2 ];
    end [shape = point, peripheries = 2 ];
    "1" [shape = box , width = 0.5, height = 0.5];
    "+1" [shape = box , width = 0.5, height = 0.5];
    "+3" [shape = box , width = 0.5, height = 0.5];
    "+4" [shape = box , width = 0.5, height = 0.5];
    "+5" [shape = box , width = 0.5, height = 0.5];

    start -> "1" [arrowhead = open , arrowtail = none];
    "1" -> "+1" [arrowhead = open , arrowtail = none];
    "1" -> "+3" [arrowhead = open , arrowtail = none];
    "+1" -> "+5" [arrowhead = open , arrowtail = none];
    "+3" -> "+4" [arrowhead = open , arrowtail = none];
    "+4" -> "+5" [arrowhead = open , arrowtail = none];
    "+4" -> end [arrowhead = open , arrowtail = none];
    "+5" -> end [arrowhead = open , arrowtail = none];
  };
  { rank = same; "+3"; "+1"; };
  { rank = same; "+5"; "+4"; };
}
```

Answer:

Question 20

Marks: 0 / 1

QE022 GraphViz-DSPL-CompilerConstruction: Change the GraphViz program below in a way that error handling is no longer part of compiler construction process.

```
digraph "Figure 10." {
  node
  "Source\nCode"      [shape=box];
  "Source\nCode"      [shape=ellipse];
  "Source\nCode"      -> "Lexical\nAnalysis";
  "Lexical\nAnalysis" -> "Syntax\nAnalysis";
  "Symbol\nTable"     [shape=diamond];
  "Lexical\nAnalysis" -> "Symbol\nTable" [arrowhead=none];
  "Error\nHandler"    [shape=diamond];
  "Lexical\nAnalysis" -> "Error\nHandler" [arrowhead=none];
  "Syntax\nAnalysis"  -> "Semantic\nAnalysis";
  "Syntax\nAnalysis"  -> "Symbol\nTable" [arrowhead=none];
  "Syntax\nAnalysis"  -> "Error\nHandler" [arrowhead=none];
  "Semantic\nAnalysis" -> "Code\nGeneration";
  "Semantic\nAnalysis" -> "Symbol\nTable" [arrowhead=none];
  "Semantic\nAnalysis" -> "Error\nHandler" [arrowhead=none];
  "Code\nGeneration"  -> "Code\nOptimization";
  "Code\nGeneration"  -> "Symbol\nTable" [arrowhead=none];
  "Code\nGeneration"  -> "Error\nHandler" [arrowhead=none];
  "Object\nCode"      [shape=ellipse];
  "Code\nOptimization" -> "Object\nCode";
  "Code\nOptimization" -> "Symbol\nTable" [arrowhead=none];
  "Code\nOptimization" -> "Error\nHandler" [arrowhead=none];
}
```

Answer:

Question 21

Marks: 0 / 1

QE032 GraphViz-DSPL-BranchingGame: Change the GraphViz program below in a way that node "dot1" is replaced by node "+6".

```
digraph "Branch 11." {
  graph [rankdir=TD];
  {
    ordering = "out";
    start      [shape      = point, peripheries = 2      ];
    end        [shape      = point, peripheries = 2      ];
    "1"        [shape      = box   , width = 0.5, height = 0.5];
    "+1"       [shape      = box   , width = 0.5, height = 0.5];
    "+3"       [shape      = box   , width = 0.5, height = 0.5];
    dot1       [shape      = point];
    "+4"       [shape      = box   , width = 0.5, height = 0.5];
    "+5"       [shape      = box   , width = 0.5, height = 0.5];

    edge [arrowhead = open , arrowtail = none];
    start -> "1" ;
    "1"   -> "+1" ;
    "1"   -> "+3" ;
    "+1"  -> dot1 ;
    "+3"  -> dot1 ;
    dot1  -> "+5" ;
    dot1  -> "+4" ;
    "+4"  -> end  ;
    "+5"  -> end  ;
  };
  { rank = same; "+1"; "+3";      };
  { rank = same; "+4"; "+5";      };
}
```

Answer:

Question 22

Marks: 0 / 1

QE032 GraphViz-DSPL-CompilerConstruction: Change the GraphViz program below in a way that Lexical, Syntax, and Semantic Analysis is replaced by Linguistic Analysis.

```
digraph "Figure 11." {
  node
  "Source\nCode"      [shape=box]          ;
  "Source\nCode"      [shape=ellipse]       ;
  "Source\nCode"      -> "Lexical\nAnalysis" ;
  "Lexical\nAnalysis" -> "Syntax\nAnalysis" ;
  "Symbol\nTable"     [shape=diamond]       ;
  "Lexical\nAnalysis" -> "Symbol\nTable" [arrowhead=none];
  "Error\nHandler"    [shape=diamond]       ;
  "Lexical\nAnalysis" -> "Error\nHandler" [arrowhead=none];
  "Syntax\nAnalysis"  -> "Semantic\nAnalysis";
  "Syntax\nAnalysis"  -> "Symbol\nTable" [arrowhead=none];
  "Syntax\nAnalysis"  -> "Error\nHandler" [arrowhead=none];
  "Semantic\nAnalysis" -> "Code\nGeneration" ;
  "Semantic\nAnalysis" -> "Symbol\nTable" [arrowhead=none];
  "Semantic\nAnalysis" -> "Error\nHandler" [arrowhead=none];
  "Code\nGeneration"  -> "Code\nOptimization";
  "Code\nGeneration"  -> "Symbol\nTable" [arrowhead=none];
  "Code\nGeneration"  -> "Error\nHandler" [arrowhead=none];
  "Object\nCode"      [shape=ellipse]       ;
  "Code\nOptimization" -> "Object\nCode" ;
  "Code\nOptimization" -> "Symbol\nTable" [arrowhead=none];
  "Code\nOptimization" -> "Error\nHandler" [arrowhead=none];
}
```

Answer:

--

End time: