Computer Science Graduation Exam 2018
Practice Questions - Programming Languages and Software Engineering

Note:

The written test of the graduation exam (July or September 2018) will consist of 60 questions which will be similar (in structure and difficulty level) with those included in this booklet. For each of the three categories (Discrete Structures and Algorithms, Programming Languages and Software Engineering, Computing Systems and Databases) there will be 20 questions.

If there are unclear aspects concerning the questions and/or answers please contact the teacher(s) who proposed the questions for each section:

C Programming Language:

- Cosmin Bonchiş (cosmin.bonchis@e-uvt.ro)

C++ Programming Language:

- Daniel Pop (daniel.pop@e-uvt.ro)
- Flavia Micotă (flavia.micota@e-uvt.ro)

Java Programming Language:

- Flavia Micotă (flavia.micota@e-uvt.ro)

Software Engineering:

- Cristina Mîndruţă (cristina.mindruta@e-uvt.ro)

1 C Language

1. The value of

   ```
   !b || (a && b)
   ```

   where `a` and `b` are variables of type `int`, is:

   (a) `TRUE` if `b` is `TRUE`
   (b) `TRUE` if `b` is `FALSE`
   (c) dependent on the value of `a`, if `b` is `TRUE`
   (d) 10 if a=4 and b=3
   (e) 1 if b is 0
   (f) 0 if a=0 and b=100

2. Given the declaration

   ```
    char c;
   ```

   the value of `c` after evaluating

   ```
    (c=getchar()) != EOF
   ```

   (a) is the value returned by `getchar()`
   (b) is 0
   (c) is 1
   (d) is 1 or 0

3. Assuming the prototype (declaration) of `square` is in sight before the call

   ```
   int a=2;
   square(a);
   ...
   ```

   and `square` is defined as:

   ```
   void square(int x){
   x=x*x;
   }
   ```

   (a) after the call, a has value 2
   (b) after the call, a has value 4

(c) the definition of `square` does not serve the purpose of squaring the actual argument in the caller function

(d) the definition of `square` is formally correct

4. What's wrong with the sequence:

```
int t[N], *low=t, *mid, *high=&t[N-1];
mid = (low+high) / 2;
```

(a) addition of two pointers is an illegal operation

(b) a pointer can't be initialized with an array

(c) the initialization of low and high is incorrect

(d) mid can't be initialized with a real value (when low+high is odd!)

5. Assuming 32-bit addresses, how many bytes of memory will be reserved by the following declarations?

```
extern long count;
struct node {
             long key;
             struct node *next;
            };
```

(a) 12 bytes

(b) 8 bytes

(c) 4 bytes

(d) none

6. Assuming the appropriate function prototypes are available, what's wrong with the sequence:

```
int *pi;
char *pc;
scanf("%d", pi);
strcpy(pc, "Timisoara");
```

(a) `pi` should be initialized before the call to `scanf`

(b) `pc` should be initialized before the call to `strcpy`

(c) the argument `pi` in the call to `scanf` should be preceded by the & operator

7. Knowing that the . and the -¿ operator have equal precedence, higher than the precedence of the * operator and assuming the following declarations, which of the expressions bellow are correct?

```
struct point {
int x, y;
};
  struct rectangle{
  struct point p1, p2;
} *r[N];
```

(a) `r[i].p1.x`
(b) `r[i]->p1.x`
(c) `(*r[i]).p1.x`
(d) `*r[i].p1.x`

8. C structures (defined with the `struct` keyword) are NOT:
   (a) aggregate types
   (b) programmer defined types
   (c) built-in types
   (d) scalar types
9. Which of the following are means of communication (sharing of data) between functions?
   (a) a function call
   (b) local variables
   (c) global variables
   (d) actual arguments
   (e) returned value
   (f) file inclusion
10. Which of the following storage classes is implicit, due to the place of the variable's declaration:
    (a) `static`
    (b) `auto`
    (c) `register`
    (d) `extern`
11. The result of executing the following two sequences of statements is:

```
seq. 1: for(exp1; exp2; exp3) statement;
seq. 2: exp1; while(exp2) statement; exp3;
```

    (a) different

   (b) identical
   (c) context-dependent

12. Can two functions, neither of whom calls the other, communicate (share data)?
    (a) no
    (b) yes, through messages
    (c) eventually, through global variables
13. If no parenthesis are used, compound expressions are evaluated:
    (a) according to the precedence of the operators
    (b) always from left to right
    (c) always from right to left
14. Given the declarations:

    ```
     static int i, t[10];
    ```

    and assuming that neither `i` nor `t` are explicitly initialized, the value of the expression

    ```
    (i==0) && (t[i]<0)
    ```

    (a) is 1
    (b) is 0
    (c) it depends on the context
15. Given the declarations

    ```
    int n=10, m=4;
    ```

    the value of the expression `1.5+n/m` is
    (a) 4
    (b) 3.5
    (c) of type `double`
    (d) of type `float`
16. Given `float x=2.5;` the value of the expression `3.0*x+10/4` is:
    (a) 10.0
    (b) 9.5

(c) of type `double`

(d) of type `float`

17. On how many bytes will `p`, declared as `void *p`, be represented?

    (a) as many as required to represent an address

    (b) 0

    (c) 1

18. How many times will the while loop below be iterated through?

    ```
    struct {
    int a:2;
    int b:5;
    } v;
    v.a=v.b=0;
    while(v.a != 3)
      v.b = 2*v.a++;
    ```

    (a) none, there are syntactic errors!

    (b) 3

    (c) it is an infinite loop!!

19. Which of the following operations are illegal?

    (a) comparison of structures

    (b) returning a structure as a function value

    (c) addition of structures

    (d) copying of a structure

20. Knowing that, by default, data of integer types are signed, which is the value of `c` after: `char c=48;`

    (a) `’`
    `0’`

    (b) `-208`

    (c) `0x30`

    (d) `48`

21. Which of the following values are logically `FALSE`?

    (a) `’0’`

   (b) `0x0`
   (c) `0`
   (d) `'\\0'`
   (e) `NULL`
   (f) `"FALSE"`

22. May a stucture have a member of the same type?
   (a) yes
   (b) no
   (c) only if it is the first member
   (d) only if it is the last member

23. Which of the following name predefined integer data types in C
   (a) char
   (b) double
   (c) short
   (d) int
   (e) byte
   (f) long

24. Which of the following operators ADMIT real operands?
   (a) increment/decrement operators
   (b) shift operators
   (c) all arithmetic operators
   (d) the logical operator NOT (!)

25. The type of `p`, used in the expression `p->m` is:
   (a) pointer to some structure
   (b) the type of `m`
   (c) void *

26. Knowing that ”0123456789” is the memory address where the string constant is generated, which is the result of evaluating the expression: ”0123456789”[i] if i=10?
   (a) the expression is syntactically wrong
   (b) undefined

(c) '
0'

(d) of type char

27. If a is an integer variable, which is the value of:

```
(a<='a') | ('z'<= a)
```

(a) 0
(b) 1
(c) it depends on the value of a

28. If a is an integer variable, which is the value of:

```
(a < 'a') && ('z' < a)
```

(a) 0
(b) 1
(c) it depends on the value of `a`

29. What amount of memory space is required to store a `struct` variable?

(a) as much as the sum of spaces required to store its members
(b) at least as much as the sum of spaces required to store its members
(c) as much as the spaces required by its largest member
(d) no space is reserved for struct variables

30. What would the value of `q-p` be if:

```
int t[20], *p=t, *q=&t[19];
```

(a) 19
(b) 20
(c) `t[19 ]- t[0]`
(d) It is illegal to substract a pointer from another pointer!

2 C++ Language

1. In C++, a constructor is a function that has the following properties:
   (a) Is a member function that has the same name like the class in which it is declared
   (b) Is a member function that returns a value
   (c) Is a member function that does not return a value
   (d) Is a member function that never has parameters
   (e) Is a member function used to initialize an object
   (f) Is a member function used to deallocate memory space
   (g) Is a member function that is used together with new operator
   (h) A class can have only one constructor
   (i) It is a method that is called by delete operator
   (j) Can be a virtual function

2. Which of the following features are supported by C++ language and not supported by C language?
   (a) Functions with default values for parameters
   (b) Reference data type
   (c) Resolution operator
   (d) Function overloading
   (e) const key word

3. If *a* and *b* are two objects of type class *Test* and the class contains a function with the following prototype: static bool equals(const Test&, const Test&);. Which is the correct call for function *equals()* from a function outside class definition?
   (a) equals(a,b);
   (b) a.equals(b);
   (c) Test.equals(a,b);
   (d) a.equals(b,a);
   (e) equals(b);

4. Which of the following remarks are true?
   (a) Operators can be overload in C++.
   (b) Java language runs into a virtual machine.
   (c) The languages Java and C++ are platform independent.
   (d) Reference variables are characteristic for languages C++ and Java.

   (e) C++ language does not have a garbage collection mechanism.
   (f) C++ language does not accept multiple inheritance.

5. Which of the following remarks are false?
   (a) In C++ and Java languages does not matter the order of member function definition.
   (b) In C++ and Java languages the comments may be specified in the same way.
   (c) In case of exception handling with try-catch mechanism in Java and C++ languages, there exists a clause that is executed in all cases (if an error is thrown or not).
   (d) In C++ and Java languages a superclass for all classes exists.
   (e) Class declaration in both C++ and Java languages finish with semicolon(;).
   (f) In C++ language all methods must to be defined inside class body.

6. Which of the following is true about templates.
   (a) Template is a feature of C++ that allows us to write one code for different data types.
   (b) We can write one function that can be used for all data types including user defined types. Like sort(), max(), min(), ..etc.
   (c) We can write one class or struct that can be used for all data types including user defined types. Like Linked List, Stack, Queue ..etc.
   (d) Template is an example of compile time polymorphism.

7. Which of the following code lines are valid in C++ language?
   (a) `const int LIGHT_ON = 0;`
   (b) `Book c = new Book("C++ Language");`
   (c) `int array[10] ;`
   (d) `final int LIGHT_OFF = 1;`
   (e) `Carte *b=null;`
   (f) `int a=9;  int *p=a;`

8. If a class X has pointer variable members, then the class should contain:
   (a) Two constructors
   (b) A empty destructor $X()\{\}$
   (c) A fiend function that copies an object of class X
   (d) = operator overloading
   (e) Copy constructor
   (f) == operator overloading
   (g) A destructor that deallocates the memory allocated for the members of type pointer

9. Which is the output of the following code sequence:

```
#include <iostream>
using namespace std;

template <typename T>
T max(T x, T y) {
    return (x > y)? x : y;
}
int main() { {
   cout << max(3, 7) << std::endl;
   cout << max(3.0, 7.0) << std::endl;
   cout << max(3, 7.0) << std::endl;
   return 0;
}
```

(a) 7
7.0
7.0

(b) Compiler Error in all cout statements as data type is not specified.

(c) Compiler Error in last cout statement as call to max is ambiguous.

(d) None of the above

10. Which of the following recommendation should be followed when an application is developed:

(a) Try to avoid, as much as possible, long function body

(b) Keep the code simple and eliminate unnecessary complexity

(c) The code should be logic and easy to understood by anyone

(d) Each code line should be comment

(e) The repeated code should be extracted in functions

(f) Separate code in modules, each module should satisfy a request

(g) Create general interfaces that serve all purposes and did not split them in little interfaces that serve specific purposes

11. Which is the correct order of constructor and destructor call in the following code?

```
class B { public: B(){}};
class M : public B { public: M(){}};
class N: protected B { public: N(){}};
class D: public N, protected M { public: D() : M(), N() {}};

int main() {
   D obj;
   return 0;
}
```

(a) Constructor: B N B M D
Destructor: B N B M D

(b) Constructor: B M B N D
Destructor: D N B M B

(c) Constructor: B N B M D
Destructor: D M B N B

(d) Constructor: D
Destructor: D

(e) Constructor: B N M D
Destructor: D M N B

(f) Constructor: B N B M D
Destructor: D N B M B

12. Which of the following assertions are true?

```
1. #include <iostream>
2. #include <vector>
3. #include <algorithm>
4. using namespace std;
5. void myfct(int i){
6.   cout << " " << i*10;
7.   }
8. int main() {
9.    vector<int> v;
10.    v.push_back(10);
11.    v.push_back(20);
12.    v.push_back(30);
13.    vector<int>::iterator it;
14.    for_each(v.begin(), v.end(),myfct);
15.    it=find(v.begin(), v.end(), 20);
16.    cout <<endl << *++it << endl;
17.    cout <<binary_search(v.begin(), v.end(), 20);
18.    return 0;
19.  }
```

   (a) After executing the line 14 the program will display 10 20 30
   (b) After executing the line 14 the program will display 100 200 300
   (c) After executing the line 14 the program will display 0 0 0
   (d) After executing the line 16 the program will display 20
   (e) After executing the line 16 the program will display 30
   (f) After executing the line 17 the program will display 2
   (g) After executing the line 17 the program will display 1
   (h) After executing the line 17 the program will display 20

13. Which is the result of the following program?

```
#include<iostream>
class Base {};
class Derived: public Base {};
int main() {
  Derived d;
  try {
    throw d;
  } catch(Base b) {
    std::cout<<"Caught Base Exception";
  } catch(Derived d) {
    std::cout<<"Caught Derived Exception";
  } catch(...) {
    std::cout<<"Other value";
  } return 0;
}
```

(a) Caught Derived Exception

(b) Caught Base Exception

(c) Compiler Error

(d) Other value

14. What is the result of the following program?

```
#include <iostream>
class Figure {
  public:
  virtual Figure* duplicate() {
    return new Figure;
  }
  virtual void display() {
    std::cout << "Figure" << std::endl;
  }
};
class Box : public Figure {
  public:
  virtual Box* duplicate() {
    return new Box;
  }
  virtual void display() {
    std::cout << "Box" << std::endl;
  }
};
int main(int argc, char** argv) {
  Figure* s1 = new Box;
  Box* b1 = s1->duplicate();
  b1->display();
  delete s1; delete b1;
  return 0;
}
```

(a) Box

(b) Figure

(c) The code does not compile

(d) The program throws a runtime exception

(e) FigureBox

(f) BoxFigure

15. The result of the following code sequence is?

```
class X {
public:
   X(int i=0) {
       p=new int;
       if(p) *p=i;
   }
   X(const X &r) {
       p=r.p;
   }
   ~X(){
       if(p)
          delete p;
   }
private:
   int *p;
};
void main() {
   X *o1=new X(1), *o2=new X(*o1);
   delete o1;
   delete o2;
}
```

(a) Compilation error.

(b) Runtime exception because a memory location is deallocated multiple times.

(c) Runtime exception because a memory location that was not allocated is dealocated.

(d) It compiles and executes without any problems.

16. What is the result of executing the following code sequence?

```
class B {
public:
  B() { cout<<"B::B()"<<endl; }
  B(const B& r) { cout << "B::B(B&)"<<endl; }
};
class A {
public:
   A(const A& r) { cout<<"A::A(A&)"<<endl; }
   A(B& bb) { cout<<"A::A(B&)"<<endl; }
};
void f(A) {
   // ...
}
int main(int, char*[]) {
   B b;
   f(b);
}
```

(a) B::B()
A::A(A&)
A::A(A&)

(b) B::B()
A::A(B&)
A::A(B&)

(c) B::B()
A::A(B&)

(d) B::B()
B::B(B&)
A::A(A&)

(e) B::B()
B::B(B&)
A::A(B&)

17. Which of the following line invokes the base constructor of Employee class in C++?

    (a) Manager::Manager(const char* s, int d) : Employee(s,d) {}

    (b) Manager::Manager(const char* s, int d) { Employee(s,d); }

    (c) Manager::Manager(const char* s, int d) { super(s,d); }?

18. What will display the following program?

```
#include <iostream>
template<class T> class Foo {
public:
 int B(T t) { return 1; }
 int B(int i) { return 2; }
 int B(int i) const { return 3; }
};
 template<> class Foo<int> {
public:
  int B(int i) { return 4; }
  int B(int i) const { return 5; }
};
 int doB() {
 Foo<int> g;
 const Foo<int> f=g;
 return f.B(10);
}
int main() {
   std::cout << doB();
  return 0;
}
```

    (a) 1

    (b) 2

    (c) 3

    (d) 4

    (e) 5

    (f) It does not compile

    (g) It does not display anything

19. If B is a public base class for class A, which of the following declaration is correct: (1) A *p = new B; or (2) B *p = new A;?

    (a) both, because the pointers to base/derived class can be initialized in both cases

    (b) none, because the pointer type does not correspond to the type of the object that it is pointing

(c) just (1)

(d) just (2)

20. Which of the following statements is true?

    (a) A static function cannot throw an exception.

    (b) A static function cannot return a non-static member of the class.

    (c) A static function cannot return a static member of the class.

    (d) A static member cannot be modified in const non-static member functions.

21. For the following declaration, how can member Counter be accessed without creating a instance of class Foo?

```
struct Foo {
 static int Counter;
 Foo(const Foo &f);
 Foo();
};
```

    (a) `Foo().Counter`

    (b) `Foo.Counter`

    (c) `Foo-> Counter`

    (d) `Foo::Counter`

    (e) `::Counter`

22. For class Foo declaration and foo() function, which of the following statements are true?

```
struct Foo {
 static int Counter;
 Foo(const Foo &f);
 Foo();
};
void foo() {
  Foo f;
  Foo f2 = f;
}
```

    (a) The copy constructor would be invoked only if f (right value) would be a constant object.

    (b) The assignment is automatically generated and it is used.

    (c) The copy constructor is called when object f2 is initialized.

23. When member functions of base class does not have sense in context of a derived class, the cause is the violation of the following OPP principle:

    (a) OCP

    (b) SRP

    (c) LSP

    (d) DRY

24. What is the result of executing the following sequence?

```
void f(int a) {
     static int x = 100;
     cout << x << " ";
     if(a>0) {
          static int y = 5;
          cout << y+x << " ";
     }
     x = 200;
     cout << x << " ";
}

void main(int, char*[]) {
     f(0);
     f(1);
}
```

(a) 100 200 200 205 200

(b) 100 200 100 105 200

(c) 100 200 100 200

(d) 100 200 200 105 200

25. Let class Employee be the base class for different types of employees of an institution. Which statement is true with regards to line labelled with (1) in the following code?

```
void f(vector<Employee*> v) {
   for(int i=0; i<v.size(); i+)
      v[i]->print(); (1)
}
// vector is a class from C++
//standard template library
```

(a) The function print() from derived class will always be called.

(b) The function print() from derived class will always be called only if the print() function is declared virtual in class Employee.

(c) The function print() from Employee class will always be called because the type of object for each it is called is Employee*.

(d) The code is not compiling because [] operator is not overloaded in class vector.

26. Exceptions are:

(a) errors that appears when a program is compiled

(b) special situation in programs resolved with the help of tests of type if(variable == NULL)

(c) errors that appears at runtime

(d) thrown using the try statement and resolved using the catch statement

(e) thrown using the throw statement and resolved using the try and catch statements

27. The OCP principle refers to:
    (a) responsibilities that a class must implement
    (b) defining a consistent class hierarchy
    (c) problems that appear because of duplicate code
    (d) the possibility of class extension and avoid the modification of existing classes and code

28. Which of the following statements are true?
    (a) In case of class templates, the compiler will generate code for all declared templates, no matter if they are instantiated or not
    (b) In case of template functions, the code will be generated just when they are called
    (c) The errors in templates declaration can be identified at compilation time
    (d) There can be errors in template declarations that are discovered just when the template is initialized
    (e) A non-type parameter of a template, e. g. $template < int\ i > \ class\ X\{\}$, can be instantiated only with constant expressions of that types (in example with constant expression that evaluates to int).

29. A virtual base class
    (a) it is instantiated only once in case of diamond-like multiple inheritance.
    (b) has all member functions are virtual
    (c) has all member functions are pure virtual
    (d) it is an abstract class inherited from a concrete class.

30. Polymorfism
    (a) refers to the mechanism of allocation/deallocation of objects in object oriented languages.
    (b) refers at the mechanism of dynamic binding at runtime
    (c) it is implemented through virtual member functions in C++
    (d) the possibility to have multiple functions with same name inside a class.

3 Java Language

1. Assume the following definitions in the Java language:

```
public interface I{
    int counter = 0;
}
public class A implements I{
     public A(){
         counter++;
    }
}
```

   Which of the following statements are true?

   (a) The value of the counter is 0 after creating an object of type A;
   (b) The value of the counter is 1 after creating an object of type A;
   (c) Incorrect interface definition as interfaces cannot have attributes;
   (d) Class A won't compile because the counter member cannot be modified;
   (e) Interface I won't compile as it cannot define attributes .

2. Assume the following definitions in the Java language:

```
public interface I1{
    int method();
    public void method1();
}
public interface I2{
    public int method();
    public void method2();
}
public abstract class C implements I1, I2{
    public int method() {
        return 0;
     }
    public void method1(){};
}
```

   Which of the following statements are true?

(a) The class C will compile;
(b) The class C won't compile because it does not completely implement the I1 interface;
(c) The class C won't compile because it does not completely implement the I2 interface;
(d) Objects of the class C cannot be created;
(e) The class C won't compile because it has no abstract methods.

3. Assume the following Java code:

```
public interface I{
}
public class C0{
        public void m(){}
}
public class C1 extends C0 implements I{
        public void m(){
                System.out.println(”m called”);
         }
}
public class C2 extends C0 implements I{
}
public class C{
        private C0[] array = new C0[]{new C1(), new C2()};
        public void m(){
                array[0].m();
                array[1].m();
         }
}
```

Which of the following statements are false?

(a) The class C won't compile;
(b) The class C2 won't compile as it has no methods;
(c) The class C benefits from polymorphism because both C1 and C2 inherits from C0;
(d) The class C benefits from polymorphism because both C1 and C2 implement I;
(e) The interface I won't compile as it has no operation defined.

4. An object is .... of a class.
   (a) a subclass
   (b) an instance
   (c) an interface
   (d) an encapsulation
   (e) a member function

5. Say that there are three classes: Computer, AppleComputer, and IBMComputer. What are the likely relationships between these classes?
   (a) Computer is the superclass, AppleComputer and IBMComputer are subclasses of Computer.
   (b) IBMComputer is the superclass, AppleComputer and Computer are subclasses of IBMComputer.
   (c) Computer, AppleComputer and IBMComputer are sibling classes.
   (d) Computer is a superclass, AppleComputer is a subclasses of Computer, and IBMComputer is a sublclas of AppleComputer

6. Consider the following Java class definition:
   ```
   abstract class C{
           abstract void m0();
           final public abstract void m1();
   }
   ```
   Which of the following statements hold?
   (a) The class will not compile as it has no method not being abstract;
   (b) The class will compile and the method m1 cannot be overridden in subclasses;
   (c) The class will not compile because of the visibility modifiers for the method m1;
   (d) The class will not compile as an abstract class has to be public;
   (e) Any non abstract class extending this class has to implement the two methods in the class C.

7. Consider the following Java class definition:
   ```
   public abstract class C{
           abstract void m0();
           protected abstract void m1();
   }
   ```
   Which of the following statements hold?

   (a) The visibility for the m1 method is in fact public because the class visibility is public;
   (b) The classes in the same package may call the m1 method;
   (c) Classes in the same package and the parent packages may call the m1 method;
   (d) The sub-classes of C may call the m1 method;
   (e) The sub-classes of C have to implement the m1 method.

8. Consider the Java code below:

```
public class C{
        private final static int MAX_LENGTH = 10;
        private Object[] numbers;
        public C(Number[] otherNumbers){
                numbers = otherNumbers;
        }
        public C(Object[] otherNumbers){
                numbers = otherNumbers;
        }
        public double computeSum(){
                int result = 0;
                for (int i = 0; i < MAX_LENGTH; i++){
                    result += (Integer)numbers[i];
                }
        return result;
        }
}
```

   Which of the following statements are true?

   (a) The first constructor always throws ClassCastException on assignment;
   (b) The computeSum may throw ArrayIndexOutOfBoundsException;
   (c) The computeSum may throw ClassCastException;
   (d) The constructors throw NullPointerException if the argument is null;
   (e) The computeSum may throw NullPointerException.

9. If a method has a reference to an immutable object, such as a String object, can the method make a change to the object?

(a) Yes, if it has a reference to an immutable object it can change it.
(b) Yes, but it must use the special "+" operator to do so.
(c) No, immutable objects can't be changed by anyone after they have been created.
(d) No, only the creator of an immutable object can change it.

10. While serializing you want some of the members not to serialize? How do you achieve it?

(a) By declaring variable static
(b) By declaring variable transient
(c) By declaring variable public
(d) None of the above

11. Consider the following Java code (Java 7 compatible):

```
public long processList1(List<Integer> list){
        long sum = 0;
        for (Integer item : list){
                sum += item;
                list.remove(item);
        }
        return sum;
}
public long processList2(List<Integer> list){
        long sum = 0;
        Iterator<Integer> it = list.iterator();
        while (it.hasNext()){
                sum += it.next();
                it.remove();
        }
        return sum;
}
```

Which of the following statements are true in relation with the Java language, in a single thread context?

(a) Both the above methods are correct implementations and they compute the same sum for the same argument;
(b) The processList2 will not compile;

(c) The processList1 may throw a ConcurrentModificationException;
(d) The processList2 may throw a ConcurrentModificationException;
(e) Both methods make the argument empty as a side effect.

12. Why will you set auto commit mode to false in case of a program that uses JDBC API?
    (a) To increase performance.
    (b) To maintain the integrity of business processes.
    (c) To use distributed transactions.
    (d) None the above.

13. Which statement from the ones below are true in relation with the Java language?
    (a) The static keyword applies to classes, attributes and methods;
    (b) A static method can only call other static methods of other objects;
    (c) A static method in a class can only refer to static attributes and methods in the same class;
    (d) A static attribute can be set only once;
    (e) A static attribute has the same value for all objects of that class.

14. Which statement from the ones below are true in relation with the Java language?
    (a) The final keyword applies to classes, attributes and methods;
    (b) A final method can only call other final methods in the same class;
    (c) A final method in a class cannot be overriden in subclasses;
    (d) A final attribute can be set only once;
    (e) Final attributes can only exist in final classes.

15. Which of the following assertions hold in relation with the Java language?
    (a) Inheritance can always be replaced by objects' composition;
    (b) A class can implement any number of interfaces;
    (c) An enumeration can implement any number of interfaces;
    (d) Using objects' composition provide a possibility to avoid access restrictions;

16. Consider the following Java definitions:

```
class A{
        public static int i=1;
        public static void main(String[] args) {
                A x=new A();
```

```
            A y=new A();
            x.i =x.i+1;
            y.i =y.i+1;
        }
}
```

Which of the following statements are true?

(a) The class will not compile;
(b) After running, x.i = 2 and y.i=2;
(c) After running, x.i = 2 and y.i=3;
(d) After running, x.i = 3 and y.i=3;
(e) After running, x.i and y.i have other values than in the above mentioned cases.

17. Which of the following Java declarations are wrong:

(a) Integer numbers= new Integer(100);
(b) int nr[]= new Integer[100];
(c) String matrix [][]= new String[2][];
(d) Integer tab= new int[100];
(e) Integer array[]= new Integer[100];

18. What can be said about the Java program below ?

```
class A {
        A(int x) { System.out.print("Constructor A called "); }
        }
class B extends A {
        B() { System.out.print("Constructor B called "); }
        }
public class Test {
        public static void main(String args[]){
            B b = new B(); }
}
```

(a) The compilation will fail;
(b) It will display : "Constructor B called"
(c) It will display: "Constructor A called"

(d) It will display: ”Constructor A called Constructor B called”

19. Which of the following statements are true in respect to the Java code below (the lines are numbered):

```
1. public class Test extends Thread{
2. public void run() {
3.         System.out.print(”Answer ”);
4.         wait(1000);
5.         System.out.print(”Question ”);
6.         }
7.         public static void main(String args []) {
8.                 Test ob = new Test();
9.                 ob.start();
10.          } }
```

(a) The compilation will fail at line 4 because wait(1000) may only be called in a synchronized block;
(b) The compilation will fail at line 4 because wait(1000) may only be called in a try/catch block ;
(c) Compilation will succeed. Nothing will be displayed on execution;
(d) Compilation will succeed. ”Answer ” will be displayed on execution;
(e) Compilation will succeed . ”Question Answer ” will be displayed on execution.

20. Which are the elements on a Java method signature:

(a) The method’s name;
(b) The method’s name and the method’s arguments’ names;
(c) The method’s name and the method’s arguments’ names and their types;
(d) The method’s name, the method’s arguments’ names and their types and the returned value’s type

21. A Java constructor:

(a) Has no type for the returned value;
(b) May be implicit;
(c) Has the same name as the class including it;
(d) Each time an object is created at least a constructor is called.

22. Which of the following Java code sequence throws an exception:

    ```
    1. public void someMethod ()
    { ...
            if ( problem ) throw new Exception(”Useful Message”);
    ...}
    2. public void someMethod () throws Exception
    { ...
            if ( problem ) Exception(”Useful Message”);
    ...}
    3. public void someMethod () throw Exception
            { ...
            if ( problem ) throws new Exception(”Useful Message”) ;
    ...}
    4. public void someMethod () throws Exception
    { ...
            if ( problem ) throw new Exception(”Useful Message”);
    ...}
    ```

    (a) 1
    (b) 2
    (c) 3
    (d) 4

23. The Java access modifiers for the members of a class are:

    (a) public, protected, default (when no modifier is specified), private
    (b) public, protected, private
    (c) public, abstract, final
    (d) public, static, final

24. Which of the statements below are false in respect to the Java language:

    (a) Any Java class has Object class like super classs
    (b) A class may inherit from one or more base classes
    (c) A class may implement one or more interfaces
    (d) Sub-classes inherit the attributes, methods and the constructors from the base class

25. What is displayed after executing the following Java code?

    String s1 = ”year” + 20 + 18, s2 = 2000 + 18 + ”year”;

    System.out.println(”s1 = ” + s1 + ”, s2 = ” + s2);

    (a) s1 = year38, s2 = 20018year
    (b) s1 = year 2018, s2 = 200018 year
    (c) s1 = year 38, s2 = 20018 year
    (d) s1 = year2018, s2 = 2018year
    (e) Runtime error: explicit cast needed

26. A Java thread is:

    (a) An instance of a class sub-classing the Thread class;
    (b) An instance of a class implementing the Runnable interface;
    (c) An object of a class having Thread asa a super-class;
    (d) An object of a class implementing the Runnable interface.

27. What is true about the String class in Java?

    (a) The content of a String cannot be modified after creation
    (b) The == operator is used to compare the content of two String objects
    (c) The String class is final
    (d) The String class has an attribute called size providing the length of the string

28. Which of the following statements is true about a Java abstract class:

    (a) A class which cannot be instantiated
    (b) A class having at least an abstract method
    (c) A class defined using the abstract keyword

29. Assume the file A.java includes the following Java definitions:

    interface X {}

    interface Y {}

    public class A implements X {}

    public class B extends A implements X, Y {}

    Why the above code will not compile?

    (a) Java does not provide multiple inheritance;
    (b) The class B double implements the X interface ;

   (c) There are two public classes in the same file;
   (d) Interfaces have to be public.

30. Assume the following Java definitions:

    ```
    class A implements Runnable {
            int counter = 0;
            public void run() {
                    while (true) counter ++;
            }
    }
    public class Test{
            public static void main(String [] arg) {
                    A a = new A();
            }
    }
    ```

    Which one of the cases below instantiate and run a thread?

    (a) run();
    (b) a.start();
    (c) new Thread(a).run();
    (d) new Thread(a).start();

4 Software Engineering

1. Check the statements which are true. A use case:
   (a) defines a function of the system.
   (b) represents what the system must do.
   (c) shows how a function of the system can be realized.
   (d) it is necessary in direct a relation with an actor.
   (e) defines the system behaviour.
   (f) defines a structure of the system.

2. Consider the following examples of requirements for an application for a medical analyses center. Check the functional requirements:
   (a) The system must allow the management of the analysis types.
   (b) The system must support maximum 20 simultaneous users.
   (c) The system must allow the update of the patient record.
   (d) The DES32 encription system must be used for the transfered data.
   (e) The system must respond in maximum 1 second to any user command.
   (f) The HELP facility must be organized hierarchically.
   (g) A history of the last 20 analysis for each patient must be available.
   (h) The sistem will be implemented in Java and will use DBMS Oracle.

3. Which statement characterizes the client-server architectural style?
   (a) Subsistems communicate using large shared data stored in a central database.
   (b) The system is composed of functional modules which process the inputs and produce outputs.
   (c) The system is made of systems that provide services and systems that request these services.
   (d) An event is broadcasted to all subsystems and is handled by the systems which are interested by this event.
   (e) Structures the system on more abstraction levels.

4. Software verification can imply
   (a) automatic static analysis
   (b) assesing usefulness and usability of the software in operational situations.
   (c) debugging
   (d) software inspections
   (e) testing to prove error existence

(f) verify that software meets user requirements

5. Check the software product types which are in CASE category.
   (a) Operating systems
   (b) Integrated development environments
   (c) Compilers
   (d) LaTeX editors
   (e) Testing tools
   (f) UML editors
   (g) Grafic editors
   (h) Software for version control
6. State machine diagram shows
   (a) system functions.
   (b) system response to internal events.
   (c) system response to external events.
   (d) interactions between objects in the system.
   (e) data structures.
   (f) interactions between actors and the system.
   (g) data flow in the system.
7. Code debugging includes
   (a) Establishing the existence of errors
   (b) Creating an hypothesis about the cause of the error
   (c) Localization of the error
   (d) Creation of acceptance tests
   (e) Error correction
   (f) Code refactoring
8. Agile methods in software development imply
   (a) Incremental delivery
   (b) Customer involvement during development process
   (c) Establishing normative processes for team working
   (d) Periodic activities to eliminate complexity from the system
   (e) Modeling the whole software before writting the code

9. Performance testing means:
   (a) a series of tests where the load is steadily increased.
   (b) the process of testing individual components in isolation.
   (c) specifications of the inputs of the test and of the expected results, also indicating the tested entity.
   (d) exercising the system beyond its maximum design load.
   (e) rerunning an existing set of tests.

10. Application frameworks are
    (a) systems developed by integrating existing application systems.
    (b) an application type generalised around a common architecture so that it can be adapted in different ways for different customers.
    (c) generic abstractions that occur across applications, represented as design patterns that show abstract and concrete objects and interactions.
    (d) collections of abstract and concrete classes that can be adapted and extended to create application systems.
    (e) shared components woven into an application at different places when the program is compiled.

11. Check the circumstances that can motivate the maintenance process.
    (a) The system does not pass one of the validation tests.
    (b) A law, in the domain for which the application was developed, changes.
    (c) Errors appear while testing the system.
    (d) Errors appear while using the system.
    (e) It is difficult to use the product, so it is requested to reorganize the elements in the user interface
    (f) A error is identified while performing a software inspection.
    (g) Customer does not validate the user interface prototype.
    (h) A component to be reused does not correspond to its specifications.
    (i) Customer decides to use another database management system.

12. Check the approaches for reusing application systems.
    (a) COTS integration
    (b) Using design patterns
    (c) Aspect oriented programming
    (d) Product line development

(e) Extension of application frameworks

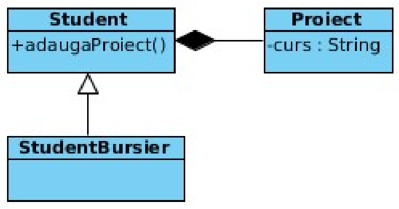
13. Version management, as activity of the software configuration management process, means

    (a) keeping track of requests for changes to the software, costs and impact of changes analysis and deciding which changes should be implemented.
    (b) keeping track of the multiple versions of system components and ensuring that changes made to components by different developers do not interfere with each other.
    (c) assembling program components, data and libraries, then compiling these to create an executable system.
    (d) preparing software for external release and keeping track of the system versions that have been released for customer use.

14. System reliability can be expresses as a verifiable requirement by

    (a) the number of transactions processed per second
    (b) memory size
    (c) rate of failure occurrence
    (d) percentage of the statements dependent on the target platform
    (e) users training time
    (f) probability of system unavailability

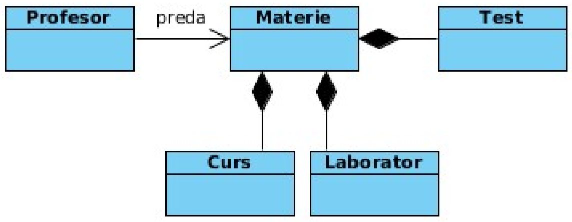
15. Consider the following class diagram.



Check the true statements?

    (a) Class `Student` inherits from class `StudentBursier`
    (b) An object of type `Student` contains a collection of objects of type `Proiect`.
    (c) Class `Proiect` has a public attribute of type `String`.
    (d) Class `Student` has the public operation `adaugaProiect`.
    (e) Class `Student` has the private operation `adaugaProiect`.
    (f) Class `Student` is superclass for the class `StudentBursier`.

16. Consider the following class diagram.



Which sequence of Java code correctly describes the relationship between classes `Profesor` and `Materie`?

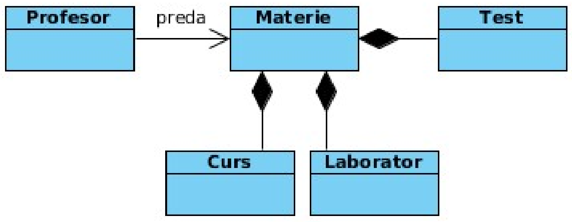
(a) `class Profesor extends Materie{...}`

(b)
```
class Profesor {
      private Materie preda; ...}
```

(c)
```
class Materie {
      private Profesor preda; ...}
```

(d)
```
class Materie {
      private Vector<Materie> preda;...}
```

17. Consider the following class diagram.



Check the true statements?

(a) Class `Materie` defines a composition of objects of type `Curs`.

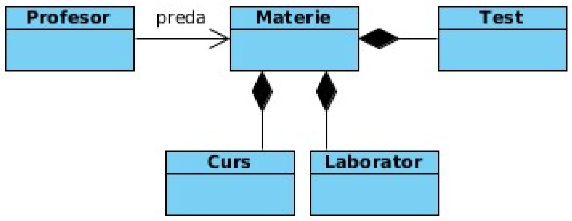
(b) A bidirectional association exists between class `Profesor` and class `Materie`.

(c) Class `Test` inherits from class `Materie`.

(d) Class `Materie` defines an aggregate of objects of type `Laborator`.

(e) An object of type `Materie` contains a collection of objects of type `Test`.

18. Consider the following class diagram.



Which sequence of Java code correctly and completely describes the relationship between class `Materie` and class `Laborator`?

(a) `class Materie extends Laborator{...}`

(b) `class Laborator extends Materie{...}`

(c)
```
class Materie {
    private Vector <Laborator> laboratoare = new Vector();...}
class Laborator {
    private Materie materie;
     ...}
```

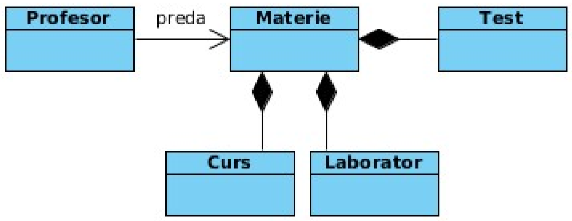
(d)
```
class Materie {
    private Laborator laborator;...}

class Laborator {
    private Vector<Materie> materie;
     ...}
```

(e)
```
class Materie {
    private Vector <Laborator> laboratoare;...}

class Laborator {
    private Materie materie;
     ...}
```

19. Consider the following class diagram.



Check the complete and correct description of the relations represented on the diagram:

(a) Association between classes `Profesor` and `Materie`; aggregation between classes `Materie`(aggregate) and `Test`(component), `Materie`(aggregate) and `Laborator`(component), `Materie`(aggregate) and `Curs`(component).

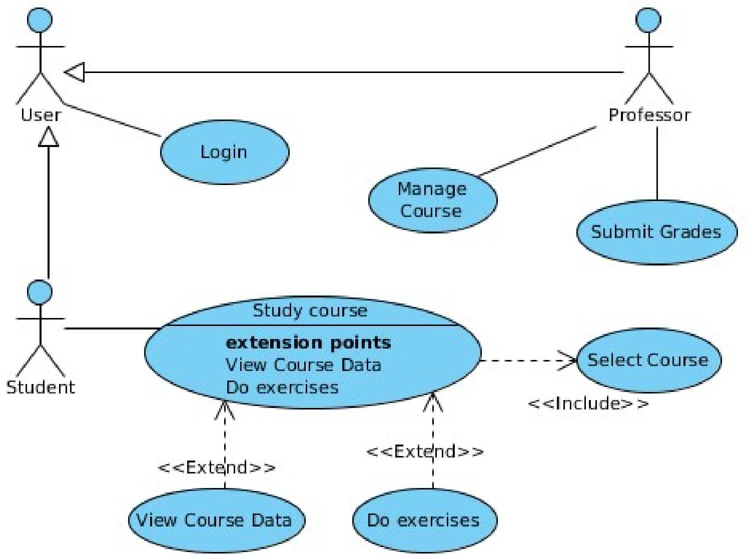
(b) Unidirectional association, named *preda*, from class `Profesor` to class `Materie`; aggregation between classes `Materie`(aggregate) and `Test`(component), `Materie`(aggregate) and `Laborator`(component), `Materie`(aggregate) and `Curs`(component).

(c) Unidirectional association, named *preda*, from class `Profesor` to class `Materie`; class `Materie` is superclass for classes `Test`, `Laborator` and `Curs`.

(d) Unidirectional association, named *preda*, from class `Profesor` to class `Materie`; composition between class `Materie`(composite) and class `Curs`(component); composition between class `Materie`(composite) and class `Laborator`(component); composition between class `Materie`(composite) and class `Test`(component).

(e) Unidirectional association, named *preda*, from class `Profesor` to class `Materie`; classes `Curs`, `Laborator` and `Test` inherit from class `Materie`.

20. Consider the following use case diagram.



Check all the use cases directly implied in the realization of the functions accesible to the student:

(a) Login

(b) Manage Course

(c) Select Course

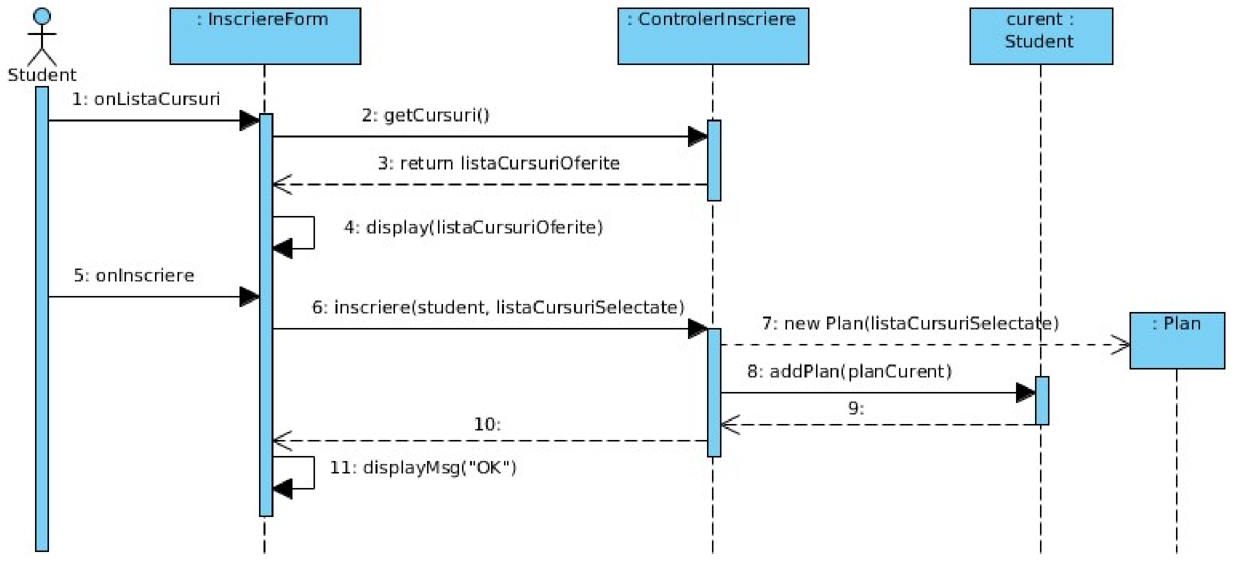
(d) Study Course

(e) View Course Data

(f) Do exercises

(g) Submit Grades

21. Consider the following sequence diagram.



Which operations of class `ControlerInscriere` result from it?

(a) `getCursuri()`

(b) `display(listaCursuriOferite)`

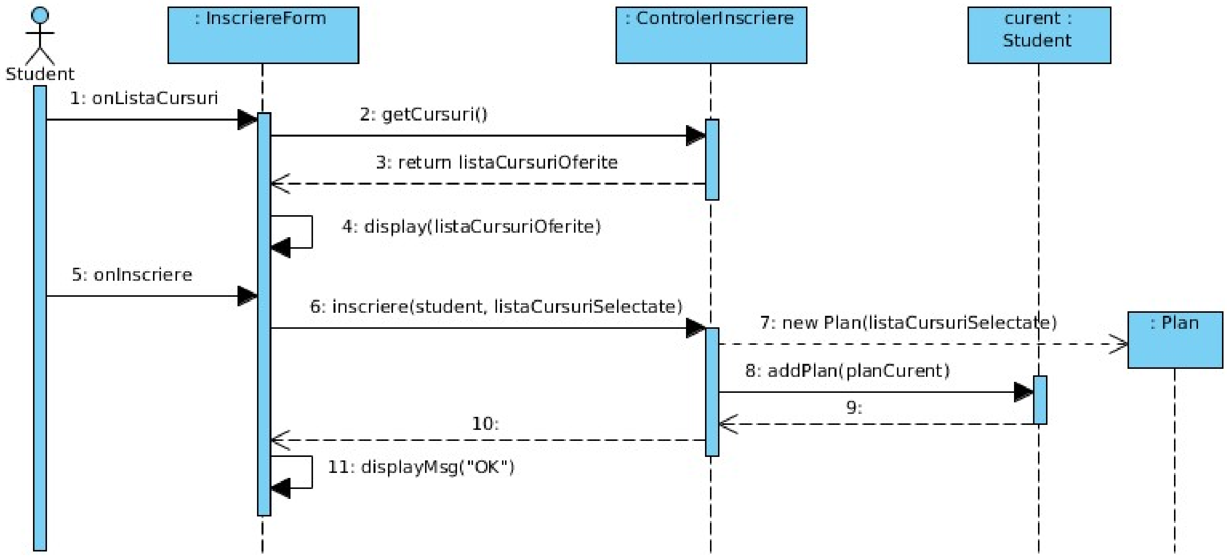
(c) `inscriere(student, listaCursuriSelectate)`

(d) `Plan(listaCursuriSelectate)`

(e) `addPlan(planCurent)`

(f) `displayMsg(OK)`

22. Consider the following sequence diagram.



Check the classes from which are instantiated the objects implied in the interaction:

(a) InscriereForm

(b) ControlerInscriere

(c) listaCursuriSelectate

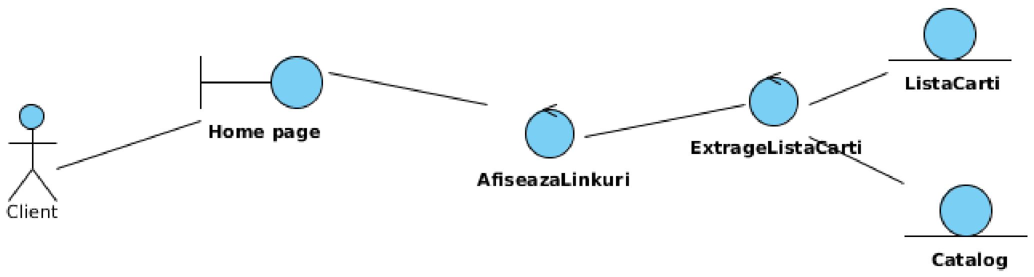
(d) curent

(e) Student

(f) Plan

(g) planCurent

23. Consider the following robustness diagram.



Which statements are true?

(a) `Home page` is a *boundary* object.

(b) `ExtrageListaCarti` can be a persistent object.

(c) `AfiseazaLinkuri` is an object for interfacing with the system.

(d) `ExtrageListaCarti` could be implemented as a method of an *entity* class.

(e) `ExtrageListaCarti` can be a persistent object.

(f) `ListaCarti` is an object for interfacing with the system.

(g) `Catalog` is an *entity* object.

24. Check the statements that comply with the rules for building robustness diagrams:

(a) Actors must communicate with *boundary* objects.

(b) Actors can communicate with *control* objects.

(c) *Boundary* objects can communicate with *control* objects.

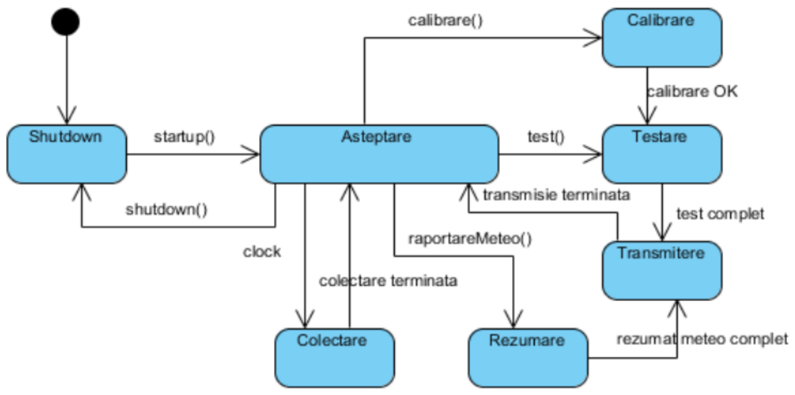
(d) *Boundary* objecte can not communicate with *entity* objects.

(e) *Entity* objects can communicate with *control* objects.

(f) *Entity* objects can communicate with actors.

(g) *Control* objects can not communicate with other *control* objects.

25. Consider the following state machine diagram.



Check the true statements.

(a) `Testare` is a state.

(b) `test()` is an internal event that triggers a transition from state `Asteptare` to state `Testare`.

(c) `test complet` is an internal event.

(d) `test complet` is a transition.

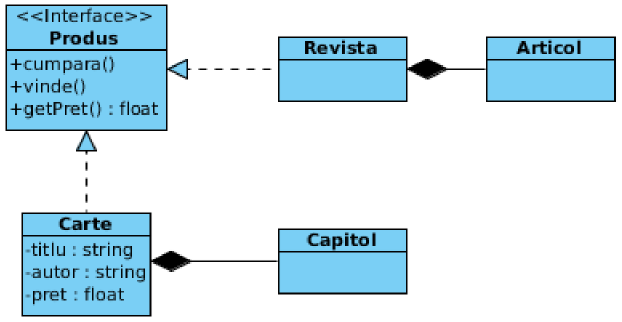
(e) Return to the state `Asteptare` after the occurrence of event `calibrare()` is realized by passing only through states `Calibrare` and `Testare`.

(f) `test complet` is an external event that triggers a transition from state `Testare` to state `Transmitere`.

(g) `test complet` is an internal event that triggers a transition from state `Transmitere` to state `Testare`.

(h) Return to the state `Asteptare` after the occurrence of event `calibrare()` is realized by passing through states `Calibrare`, `Testare` and `Transmitere`.

26. Consider the following class diagram.



Which sequences of Java code are valid for class `Carte`?

(a) `class Carte extends Produs{...}`

(b) `private float pret;`

(c) `private pret float;`

(d) `public float getPret();`

(e) `class Carte implements Produs{...}`

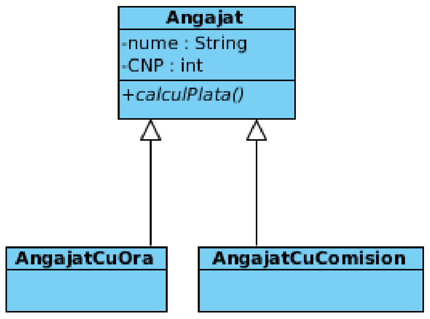
(f) `private float pret(){...}`

(g) `public float getPret(){...}`

(h) `public Produs cumpara(){...}`

(i) `private Vector<Capitol> capitole = new Vector();`

27. Consider the following class diagram.



Which sequence of Java code correctly and completly defines what results from the diagram for class `Angajat`?

(a)
```
class Angajat {
      private String nume;
```

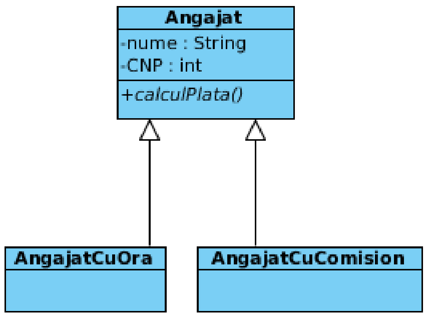
```
        private int CNP;
        public abstract void calculPlata();
         ...}
```

(b)
```
abstract class Angajat {
        private String nume;
        private int CNP;
        public abstract void calculPlata();
         ...}
```

(c)
```
abstract class Angajat {
        private String nume;
        private int CNP;
        public abstract void calculPlata(){...};
         ...}
```

(d)
```
abstract class Angajat {
        private String nume;
        private int CNP;
        public void calculPlata();
         ...}
```

28. Consider the following class diagram.



Which sequences of Java code are valid for class `AngajatCuOra`?

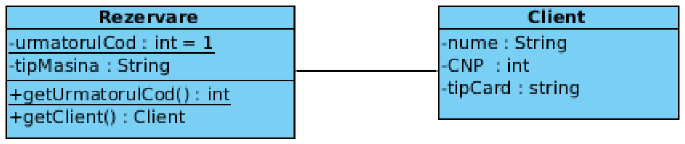
(a) `class AngajatCuOra extends Angajat{...}`

(b) `class AngajatCuOra implements Angajat{...}`

(c) `public calculPlata();`

(d) `public calculPlata(){...};`

29. Which sequences of Java code are valid for class `Rezervare`?



(a) `class Rezervare extends Client{...}`

(b) `private int urmatorulCod = 1;`

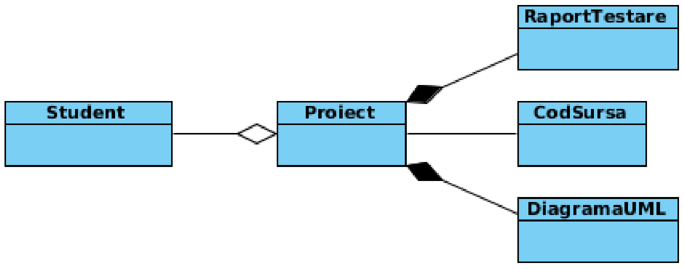
(c) `private static int urmatorulCod = 1;`

(d) `public static int getUrmatorulCod();`

(e) `private Client client;`

(f) `public static Client getClient();`

30. Consider the following class diagram.



Select the true statements.

(a) Class `Student` defines an aggregate of objects of type `Proiect`; classes `RaportTestare` and `DiagrameUML` define compositions of objects of type `Proiect`.

(b) Class `Proiect` defines compositions of objects of type `Student`, of objects of type `DiagrameUML` and of objects of type `RaportTestare`.

(c) Class `Proiect` defines an aggregate of objects of type `Student` and compositions of objects of type `DiagrameUML` and of objects of type `RaportTestare`.

(d) Class `Proiect` defines a composition of objects of type `Student` and aggregates of objects of type `DiagrameUML` and of objects of type `RaportTestare`.

(e) Class `Proiect` has an association relationship with class `CodSursa`.

5 Answers

C Language

1. 1b
2. 2d
3. 3a,3c,3d
4. 4a
5. 5d
6. 6a,6b
7. 7b,7c
8. 8c, 8d
9. 9c, 9d, 9e
10. 10a,10b
11. 11a
12. 12c
13. 13a
14. 14a
15. 15b, 15c
16. 16b, 16c
17. 17a
18. 18c
19. 19b,19c
20. 20c, 20d
21. 21b, 21c, 21e
22. 22b
23. 23a, 23c, 23d, 23f
24. 24c, 24d
25. 25a
26. 26c, 26d
27. 27c
28. 28a
29. 29b
30. 30a

C++ Language

1. 1a,1c,1e,1g
2. 2a,2b,2c,2d
3. 3c,3d
4. 4a,4b,4d,4e
5. 5c,5d,5e,5f
6. 6a,6b,6c,6d
7. 7a,7c
8. 8d,8e,8g
9. 9c
10. 10a,10b,10c,10e,10f
11. 11c
12. 12b,12e,12g
13. 13b
14. 14c
15. 15b
16. 16c
17. 17a
18. 18e
19. 19d
20. 20b
21. 21d
22. 22c
23. 23c
24. 24a
25. 25b
26. 26c,26e
27. 27d
28. 28b,28d,28e
29. 29a
30. 30b,30c

Java Language

1. 1d
2. 2a,2d
3. 3a,3b,3d,3e
4. 4b
5. 5a
6. 6c
7. 7b,7d
8. 8b,8c,8e
9. 9c
10. 10a,10b
11. 11c
12. 12a,12b,12c
13. 13a,13c,13e
14. 14a,14c,14d
15. 15b,15c
16. 16d
17. 17b,17d
18. 18a
19. 19b
20. 20d
21. 21a,21b,21c,21d
22. 22d
23. 23a
24. 24b,24d
25. 25d
26. 26a,26c
27. 27a,27c
28. 28a,28c
29. 29c
30. 30d

Software Engineering

1. 1a,1b
2. 2a,2c,2g
3. 3c
4. 4a,4d,4e
5. 5b,5c,5e,5f,5h
6. 6b,6c
7. 7b,7c,7e
8. 8a,8b,8d
9. 9a
10. 10d
11. 11b,11d,11e,11i
12. 12a,12d
13. 13b
14. 14c,14f
15. 15b,15d,15f
16. 16b
17. 17a,17d,17e
18. 18c
19. 19d
20. 20a,20c,20d,20e,20f
21. 21a,21c
22. 22a,22b,22e,22f
23. 23a,23d,23g
24. 24a,24c,24d,24e
25. 25a,25c,25h
26. 26b,26e,26g,26i
27. 27b
28. 28a,28d
29. 29c,29d,29e
30. 30c,30e