1.

Question 1 of 25

It considers the following source code:

```
#include <iostream>
#include <string.h>
using namespace std;

class Person {
public:
     int age;
     char* name;
     Person(int v=0, char* n="Name"):age(v){
          this->name = new char[strlen(n)+1];
          strcpy(this->name,n);
     }
     Person(Person& p){
          this->age = p.age;
          this->name = new char[strlen(p.name)+1];
          strcpy(this->name, p.name);
     }
     ~Person(){
          delete [] this->name;
     }
     void operator=(Person& p){
          this->age = p.age;
          delete[] this->name;
          this->name = new char[strlen(p.name)+1];
          strcpy(this->name, p.name);
     }

     Person operator+(Person p, int v){
          Person t;
          t.age = this->age+p.age+v;
          return t;
     }
};

Person Medie(Person a, Person b){
     Person p;
     p.age=(a.age+b.age)/2;
     return p;
}

void main(){
     Person p1, p2(20, "John");
     Person p3;
     p3 = p1+p2+10;
}
```

Which of the following statements is CORRECT?

- A. A run-time error is generated because of the operator = :

Which of the following statements is CORRECT?

A. A run-time error is generated because of the operator = ;

B. A compile-time error is generated because the operator + overloaded method is not called properly;

C. A run-time error is generated because of the copy constructor;

A compile-time error is generated because of number of parameters of the operator + overloaded method;

E. The application runs properly and the age attribute of the object p3 is modified;

Reset Selection

D, deoarece operator + nu a fost definit corespunzator.
Exemplu operator corect :Complex operator +(const Complex& src);

2.

Question 2 of 25

It considers the following source code:

```
#include <iostream>
#include <string.h>
using namespace std;

class Car{
private:
     int prodDate;
     char* color;
public:
     Car(){
          prodDate=20120704;
          color=0;
     }
     Car(int an = 0, char* cul = ""){
          this->prodDate = an;
          this->color = new char[strlen(cul)+1];
          strcpy(this->color, cul);
     }
     ~Car(){
          delete[] this->color;
     }
};
void main(){
     Car carA, carB;
}
```

Which of the following statements is CORRECT?

A. The default constructor method is called 2 times;

B. The attributes are defined in a wrong way;

C. A run-time error is generated;

D. The destructor method is defined in a wrong way;

A compile-time error is generated;

Reset Selection

Raspuns: E
Sunt doi constructori default si nu stim pe care il apeleaza - compile error:
Car() { } si Car (int an = 0, char* cul = "")
Dc al doilea constructor nu avea parametri initializati, programul ar fi rulat.

3.

Question 3 of 25 2.0 Points

It considers the following OpenSSL function call RSA_private_encrypt(sizeof(finalDigest), buf, e_data, apriv, RSA_PKCS1_PADDING). The private key has a length of 1024 bits. Which of the following statements is CORRECT?

A. RSA_PKCS1_PADDING is a parameter specifying the padding length;

B. sizeof(finalDigest) represents the digest of the message;

C. A compile-time error is generated because the function has other header;

RSA_PKCS1_PADDING is a parameter specifying the padding type;

E. RSA_PKCS1_PADDING has the same length with the private key;

Reset Selection

Raspuns : D
Conform standardului de openssl padding este un integer ptr a specifica tipul de padding.
https://www.openssl.org/docs/crypto/RSA_private_encrypt.html

4.

Question 4 of 25

It considers the following source code:

```
#include <stdio.h>
#define NMAX 100
#define then
#define BEGIN {
#define END }
#define INTEGER

int void main()
BEGIN
INTEGER S = 0;
INTEGER vector[NMAX];
for(INTEGER i=0; i<NMAX; i++)
     vector[i]=i+1;
for(INTEGER i=0; i<NMAX; i++)
     S+=vector[i];
printf("S= %d",S);
END
```

At run-time, the result of the above application is:

A. The application generates a run-time error;

B. 5050;

The source code cannot be compiled;

D. 50;

E. 0;

Reset Selection

Raspuns : C

Avem tip de return int si void in acelasi timp. Nu se poate compila.
Daca s-ar sterge int si s-ar adauga int, ca in exemplul de mai jos:

```
# define INTEGER int
```

programul ar rula si am obtine valoarea 5050.

5.

Question 5 of 25

It considers the following source code:

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

class Polygon{
public:
     virtual void f(){cout<<"Polygon class"<<endl;}
};

class Rectangle:public Polygon{
public:
     void f(){cout<<"Rectangle class"<<endl;}
};

class Triangle:public Polygon{
public:
     virtual void f(){cout<<"Triangle class"<<endl;}
};

void main(){
     Polygon p, *pp;
     Rectangle r, *pr;
     Triangle t, *pt;
     pp=&p;
     pr=&r;
     pt=&t;
     p=r;
     p.f();
     pp=pr;
     pp->f();
     pp=pt;
     pp->f();
}
```

Which of the following statements is CORRECT?

A. The program displays the strings: Polygon class, Polygon class, Triangle class;

B. The program displays the strings: Polygon class, Polygon class, Polygon class;

C. A compile-time error is generated because the conversion derived object to base object is not allowed;

The program displays the strings: Polygon class, Rectangle class, Triangle class;

E. A run-time error is generated because the methods f are not declared as virtual in all defined classes;

Reset Selection

Raspuns: D
p=r - obiectul r a fost copiat in obiectul de tip p si s-a facut downcast la clasa de baza. Apelarea functiei ne va da clasa de baza.
pp = pr - pp si pr sunt pointeri si nu obiecte, iar atribuirea lui pr la pp a functionat datorita derivarii. Apelarea functiei f() afiseaza Rectanble si nu Polygon, datorita pointerului si datorita faptului ca metoda este virtuala in clasa de baza. Altfel ar fi aparut tot Polygon pe ecran.

6.

Question 6 of 25

In C++, a static attribute declared in a class is:

Used by class to manage the object collection;

B. Always initialized in class definition;

C. A data definition when it is declared;

D. Always defined as private attribute;

E. A member for each object having the class as data type;

Reset Selection

Raspuns : A
Nu este initializa in definerea clasei.
Poate fi definit ca atribut public
Poate avea mai multe tipuri(int, double ...).

7.

```
#include <string.h>
using namespace std;

class Person {
public:
     int age;
     char* name;
     Person(int v=0, char* n="Name"):age(v){
          this->name = new char[strlen(n)+1];
          strcpy(this->name,n);
     }
     Person(Person& p){
          this->age = p.age;
          this->name = new char[strlen(p.name)+1];
          strcpy(this->name, p.name);
     }
     ~Person(){
          delete [] this->name;
     }
     void operator=(Person& p){
          this->age = p.age;
          delete[] this->name;
          this->name = new char[strlen(p.name)+1];
          strcpy(this->name, p.name);
     }
};

Person Medie(Person a, Person b){
     Person p;
     p.age=(a.age+b.age)/2;
     return p;
}

void main(){
     Person p1, p2(20, "John");
     Person p3;
     p3 = Medie(p1, p2);
}
```

Which of the following statements is CORRECT?

A. Copy constructor is called 2 times;

B. A compile-time error is generated because of ambiguity of constructor methods;

C. A run-time error is generated because of destructor method;

. Copy constructor method is called 3 times;

E. Operator = is called 2 times;

Reset Selection

Raspuns: D - In calculul Mediei se apeleaza constructorul de copiere de 3 ori deoarece **parametri se trimit prin valoare(2 copiere + noul copierea noului obiect la return)**

8.

Question 8 of 25

It considers the following source code:

```
#include <iostream>
#include <string.h>
using namespace std;

class Car{
private:
     int prodDate;
     char* color;
public:
     Car(){
          prodDate=20120704;
          color=0;
     }
     Car(int an, char* cul){
          this->prodDate = an;
          this->color = new char[strlen(cul)+1];
          strcpy(this->color, cul);
     }
     ~Car(){
          delete[] this->color;
     }
};

void main(){
     Car carA, carB(20120615, "Red");
     Car carC = carB;
}
```

Which of the following statements is CORRECT?

A. A run-time error is generated because of default constructor method

B. A compile-time error is generated because of object defining;

C. The application runs properly;

D. A compile-time error is generated because of object assignment;

E. A run-time error is generated because of destructor method;

Reset Selection

Raspuns : C
Aplicatia ruleaza corect deoarece avem un singur constructur default.
Similara cu intrebarea nr. 2.

9.

Question 9 of 25

It considers the following source code:

```
#include <vector>
#include <list>
#include <algorithm>
#include <iostream>
using namespace std;

void main(){
     vector<int> v1;
     int dim=20;
     int x;
     v1.reserve(dim);
     for (int i=0; i < dim; i++) {
          x = (i+1)*10;
          v1.push_back(x);
     }

     list<int> l;
     list<int>::iterator itl;
     for(int i=0; i<10;i++)
          l.push_back(v1[i]);
     for(int i=10; i<dim; i++)
          l.insert(l.begin(), v1[i]);
     for (itl=l.begin(); itl != l.end(); itl++)
          cout<<(*itl)<<endl;
}
```

Which of the following statements is CORRECT?

A. The application displays 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200;

The application displays 200, 190, 180, 170, 160, 150, 140, 130, 120, 110, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100;

C. A compile-time error is generated;

D. The application displays 200, 190, 180, 170, 160, 150, 140, 130, 120, 110, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10;

E. A run-time error is generated;

Reset Selection

Raspuns : B

Atentie la iteratii. In prima iteratie se pun in vectorul v valorile 10,20,30 ... 200

In iteratia cu lista pun prima oara valorile vectorului 10, 20 .. pana la 100. Apoi se insereaza in ordine inversa valorile 110, 120 ... pana la 200.

Apoi se citeste lista. Push_back - pune elementul la final, l.begin - la inceputul stivei

10.

Question 10 of 25

In ASN1 V3 certificate structure, the OpenSSL function X509_set_version setting the X509 certificate version:

A. Writes value 2 in the first byte of certificate;

Writes value 2 in an INTEGER ASN1 Data Type;

C. Writes value 3 in a SEQUENCE ASN1 Data Type;

D. Writes value 3 in an INTEGER ASN1 Data Type;

E. Writes value 3 in an OBJECT IDENTIFIER ASN1 Data Type;

Reset Selection

Raspuns B

Valoarea scrisa este 2 si este de tip integer.

`Vezi aici; ftp://www.ietf.org/rfc/rfc3280.txt -> 4.1  Basic Certificate Fields`

11.

Question 11 of 25

For the next code sequence

```
2:
3: class Student implements Person {
4: public void Speak(){}
5: }
6:
7: abstract class MasterStudent extends Student {}
8:
9: abstract class PhDStudent extends Student {
10: public void Speak(String message){}
11: }
12:
13: class Graduate extends Student implements Person{
14: public void Speak(){}
15: }
```

what is the result ?

A. Compilation succeeds

Compilation error at line 3

C. Compilation error at line 7

D. Compilation error at line 9

E. Compilation error at line 13

Reset Selection

Raspuns B
Dc pe linia 1 am fi avut codul urmator : public interface Person {} , programul ar compila cu succes.
Dar in acest caz vom primi eroare la linia 3 intrucat interfata Person nu exista.
Extends - ptr clase
Implements - ptr interfete

12.

Question 12 of 25

Given:

```
public static void main(String[] args) {
   double[] frequency = {10.5, 3.45, 15.99, 3.99};
   int[] values = {5, 8, 11, 29};
   try {
      DataOutputStream out = new DataOutputStream(new BufferedOutputStream(new FileOutputStream("test.txt")));
      for (int i = 0; i < frequency.length; i++) {
         out.writeDouble(frequency[i]);
         out.writeInt(values[i]);
      }
      out.flush();
      DataInputStream in = new DataInputStream(new BufferedInputStream(new FileInputStream("test.txt")));
      try {
         while (true) {
            double frecv = in.readDouble();
            int val = in.readInt();
            System.out.format("Value %d - %.2f frequency ", val, frecv);
         }
      } catch (EOFException e) {
         //aici iese cand s-a terminat fisierul
      }
   } catch (IOException ioe) {
      ioe.printStackTrace();
   }
}
```

A. Value 1076166656 - 0,00 frequency Value 1074502041 - -0,00 frequency Value 1076886241 - 19991590273549737000000000000000000000,00 frequency Value 1074785157 - 0,00 frequency

B. the example generates runtime exceptions

A. Value 1076166656 - 0,00 frequency Value 1074502041 - -0,00 frequency Value 1076886241 - 19991590273549737000000000000000000000,00 frequency Value 1074785157 - 0,00 frequency

B. the example generates runtime exceptions

C. Value 5 - 10,50 frequency Value 8 - 3,45 frequency Value 11 - 15,99 frequency Value 29 - 3,99 frequency

D. the example generates compiler errors

E. the example does't print something because the text file is empty

Raspuns C:

Atentie la ordinea de citire si scrire a datalor. Dc se scrie float si se citeste int, atunci raspunsul este A. In cazul nostru atat scrierea cat si citirea sunt in aceeasi ordine.

13.

Question 13 of 25

Being given the next class

```
public class Box {
	Object value;

	public Box(Object value){
		this.value = value;
	}

	public Object getValue(){
		return this.value;
		}

	public void setValue(Object value){
		this.value = value;
	}
}
```

and the test

```
		Box[] numericalValues = new Box[3];
		numericalValues[0] = new Box(10);
		numericalValues[1] = new Box(20);
		numericalValues[2] = new Box("30");

		int sum = 0;
		for(Box box : numericalValues)
			sum += (int)box.getValue();

		System.out.println("The sum is:"+sum);
```

select the correct affirmation:

A. The test will generate a compiler error on line numericalValues[2] = new Box("30");

B. The test will run without problems and it will print "The sum is 60"

The test will generate a compiler error on line sum += (int)box.getValue();

D. The test will generate a compiler error on line numericalValues[0] = new Box(10);

E. The test will generate a run-time exception on line sum += (int)box.getValue();

Reset Selection

Raspuns C:
Genereaza compilation error deoarece obiectului box nu i se poate face cast la integer.
Compilation error = atunci cand eroare este generata de scrierea codului gresi
Runtime error = codul se poate compila, dar este implementat incorect algoritmul. Codul se compileaza. Sunt bugurile din programe ...
new Box[3] - aloca memorie ptr 3 obiecte de tip Box, obiectele nu au fost create

14.

Question 14 of 25

The next sequence

```
String name1 = "John";
String name2 ="John";

if(name1 == name2)
System.out.println("Strings are equal");
else
System.out.println("Strings are NOT equal");

String name3 = new String("John");
if(name1 == name3)
System.out.println("Strings are equal");
else
System.out.println("Strings are NOT equal");

Integer i1 = 10;
Integer i2 = 10;

if(i1 == i2)
System.out.println("Integers are equal");
else
System.out.println("Integers are NOT equal");

Integer i3 = 300;
Integer i4 = 300;
if(i3 == i4)
System.out.println("Integers are equal");
else
System.out.println("Integers are NOT equal");
```

will print:

Strings are equal
Strings are NOT equal
Integers are equal
Integers are NOT equal

B. Strings are equal
Strings are equal
Integers are equal
Integers are equal

C. Strings are equal
Strings are NOT equal
Integers are equal
Integers are equal

D. Strings are equal
Strings are equal
Integers are equal
Integers are NOT equal

E. Strings are NOT equal
Strings are equal
Integers are equal

Raspuns A:
Primele doua stringuri create au aceeasi adresa de memorie. E ca si cum am crea 2 referinte ptr aceiasi valoare. Cand se foloseste new se creaza un nou obiect, deci o noua adresa de memorie.
Ptr Integers(nu este similar cu int) daca valoarea depaseste este mai mare ca 127 avem valori diferite de memorie, deci i3 si i4 nu sunt egale.
In ambele situatii se compara adresele obiectelor si nu valorile.

15.

Question 15 of 25

In the next sequence

```
class Something{
        Integer value = 200;
        Something doSomething(Something s){
            s = null;
            return s;
        }
        public static void main(){
            Something s1 = new Something();
            Something s2 = new Something();
            Something s3 = s1.doSomething(s2);
            s1 = null;

            //other
        }
 }
```

How many objects are eligible for GC when line //other is reached?

A. 2

B. 3

C. 1

D. Impossible to determine

E. 0

Reset Selection

Raspuns A

Sunt doua obiecte deoarece obiectului s1 i se atribuie valoarea null si obiectului s3 i se atribuie valoarea dupa apelul metodei doSomething, avand valoarea *null*.

```
Valorile sunt s1:null s2:ism.ase.ro.Something@a32b s3:null
```

16.

Question 16 of 25

In the next code sequence

```
public class Test {
public static void main(String [] args){
doSomething(1);
doSomething(1,2);
}
    //insert here method definition
}
```

which of the following code blocks can be inserted independently without compile errors (Choose all that apply)

A. static void doSomething(int... args, int a){ }

B. static void doSomething(int[] args){ }

C. static void doSomething(int... args){ }

D. static void doSomething(int a, int... args){ }

E. static void doSomething(int args...){ }

Raspuns : C, D
int... args - permite sa avem nr de parametri variabili. Atentie ca acest tip de parametru trebuie sa fie ultimul din lista de parametri din cadrul metodei(din acest motiv A nu este corect).

int[] args - nu functioneaza deoarece avem nevoie de un vector
int args ... - sintaxa este incorecta.

17.

Question 17 of 25

Given

```
class Base{
   static { System.out.print("b1 ");}      //static init block
   { System.out.print("b2 ");}            //instance init block
   public Base(){
      System.out.print("b3 ");
   }
}
class Subclass extends Base{
   public Subclass(){
      System.out.print("s1 ");
   }
}
public class Main {
   public static void main(String[] args) {
      System.out.print("main ");
      new Base();
      new Subclass();
   }
}
```

what is the result ?

A. b1 main b2 b3 s1 b2 b3

B. main b1 b2 b3 b2 b3 s1

C. main b2 b3 s1 b1 b2 b3

D. main b1 s1 b2 b3

E. main b1 b3 s1 b2 b3

F. It can't be determined

G. main b1 b2 b3 s1 b2 b3

Reset Selection

Raspuns : B

Se executa blocul main.

Se creaza obiectul de baza, in care se initializeaza doar o data static.

Se creaza obiectul subclasa, la care se adauga si constructorul de baza, cu exceptia elementului static.

```
main b1 b2 b3 b2 b3 s1
```

18.

Question 18 of 25

```
class Parent{
    public Parent(String s){
        System.out.print("P");
    }
}
public class Child extends Parent{
    public Child(String s){System.out.print("C");}
    public static void main(String[] args){
        new Child("2");
    }
}
```

A. PC2
B. PC
C. 2
Compilation error
E. CP
F. CP2

Reset Selection

Raspuns : D
Nu exista constructorul default in clasa de baza. Dc ar fi definit public Base() { } rezultatul ar fi PC. Ar mai fi functionat daca aveam public Child(String super (s))

19.

Raspuns:

Question 19 of 25

Which one from the next Java statements it CORRECT:

A. interfaces can contain instance variables
B. abstract classes can be instantiated in objects;
C. All these statements are NOT correct in Java
D. abstract classes can contain non-abstract methods and instance variables;
E. interfaces can NOT be used as reference type;

Reset Selection

D

Clasele abstracte nu pot fi instantiate.
Similar C++, clase care contine cel putin o metoda virtuala.

20.

Question 20 of 25

For the next sequence

```
class Foo{
Integer code;
String description;
int[] values;
public Foo(int code, String description, int no){
this.code = code;
this.description = description;
for(int i = 0;i<no;i++)
values[i] = i+1;
}
}

public class Test {
public static void main(String[] args) {
Foo[] foos = new Foo[5];
System.out.println("Done !");
}
}
```

How many objects are created before printing Done!

A. 20

B. 6

C. 21

D. 15

1

5

G. 0

H. 15

I. 16

Reset Selection

Raspuns : E si nu F

Foo[] foos = new Foo[5]; se aloca memorie ptr 5 obiecte dar care sunt neinitializate. In schimb se creaza un obiect de tip array. Deci avem un singur obiect creat.

21.

Question 21 of 25

The next Java example:

```
class Base{
public int vb;
public void Test(){System.out.print(" Test 1");}
}
class Subclass extends Base{
public void Test(){System.out.print(" Test 2");}
}
public class Main {
    public static void main(String[] args) {
                Base b = new Base();
                b.Test();
                Subclass d = new Subclass();
                d.Test();
                d = (Subclass) b;
                d.Test();

    }
}
```

prints:

A. prints Test 1 Test 2 Test 2

B. prints Test 2 Test 2 Test 1

C. prints Test 1 Test 2 Test 1

the example generates a runtime ClassCastException because you can NOT do Downcasting

E. prints Test 1 Test 1 Test 1

Reset Selection

Raspuns D

La linia d=(Subclass) b - se incearca sa se transforme o clasa de baza in clasa derivata. Acest lucru nu este posibil. Daca era invers se putea. Adica din clasa derivata sa fie transformata in clasa de baza.

```
Base c = new Base();
c = (Base) d;
c.Test();
```

! Atentie ca aici se va apela tot metoda clasei derivate

22.

Question 22 of 25

The next Java example

```
class Vector {
    public int[] valori;
    public Vector(int n){
        valori = new int[n];
        for(int i=0;i<n;i++)
                valori[i] = i+1;
    }
    public void Print()
    {
        System.out.print(" \n Values:");
        for(int vb : valori){
            System.out.printf(" %d", vb);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Vector vv1 = new Vector(3);
        Vector vv2 = new Vector(5);

        vv2 = vv1;
        vv1.valori[0] = 1000;
        vv1.Print();
        vv2.Print();
    }
}
```

prints at console:

. Values: 1000 2 3
Values: 1000 2 3

B. Values: 1 2 3 4 5
Values: 1 2 3 4 5

C. Values: 1000 2 3
Values: 1 2 3 4 5

D. Values: 1000 2 3 4 5
Values: 1000 2 3 4 5

E. Values: 1000 2 3
Values: 1000 2 3 4 5

Reset Selection

Raspuns A
Valorile vectorilor la initializare sunt
1,2,3 si respectiv 1,2,3,4,5.
Apoi se modifica primul index din vector vv1 din 1 in 1000.
La linia vv2 = vv1, vv2 va contine adresa de memorie a lui vv1.
La apelarea funtiei print ambele for afisa acelasi lucru.

23.

Question 23 of 25

Given:

```
class Base{
    static { System.out.print("b1 ");}       //static init block
    { System.out.print("b2 ");}              //instance init block
    public Base(){
        System.out.print("b3 ");
    }
}
class Subclass extends Base{
    public Subclass(){
        System.out.print("s1 ");
    }
}
public class Main {
    public static void main(String[] args) {
         System.out.print("main ");
        new Subclass();
        new Base();
    }
}
```

what is the result ?

A. It can't be determined

B. main s1 b1 b2 b3

C. main b2 b3 s1 b1 b2 b3

D. b1 main b2 b3 s1 b2 b3

E. main b1 b2 b3 s1 b2 b3

F. main b1 b3 s1 b2 b3

G. main b1 s1 b2 b3

Reset Selection

Raspuns D
Intrebarea este similara cu nr. 17, doar ca aici se schimba ordinea.
Main, parametru static, subclasa(care apeleaza constructor baza si apoi constructor subclasa), constructor clasa de baza

24.

Question 24 of 25

The next Java program that uses Integer objects:

```
public static void Modify(Integer a, Integer b, Integer c){
         b = 20;
         c = 30;
         a = c;
 }
 public static void main(String[] args) {
        Integer a = 10, b = 2, c = 3;
        Modify(a, b, c);
        System.out.println("a = "+a+" and b = "+b+" and c ="+c);

}
```

generates the results:

a = 10 and b = 2 and c = 3

B. a = 10 and b = 20 and c = 3;

C. the source code has compiler errors because the variables a,b,c are modified inside the method.

D. a = 30 and b = 20 and c = 30;

E. a = 10 and b = 20 and c = 30;

Reset Selection

Raspuns : A
Valorile a, b, c se modifica in interiorul functiei, dar cele initiale nu se modifica dupa terminarea apelului. Deci vom avea aceleasi valori ca cele din linia
Integer a =10, b =2, c=3;

25.

Question 25 of 25

Which one from the next Java statements it is correct:

A. abstract classes can NOT be instantiated in objects;

B. abstract classes can contain non-abstract methods and instance variables;

C. classes containing abstract methods are called abstract classes;

D. an abstract class is defined using abstract keyword

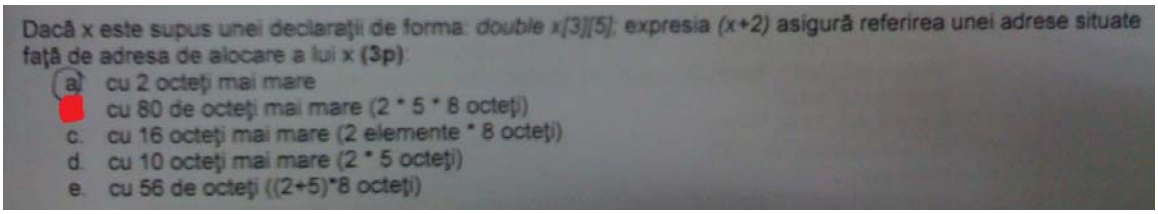
. all these statements are correct in Java

F. interfaces can NOT contain instance variables

Reset Selection

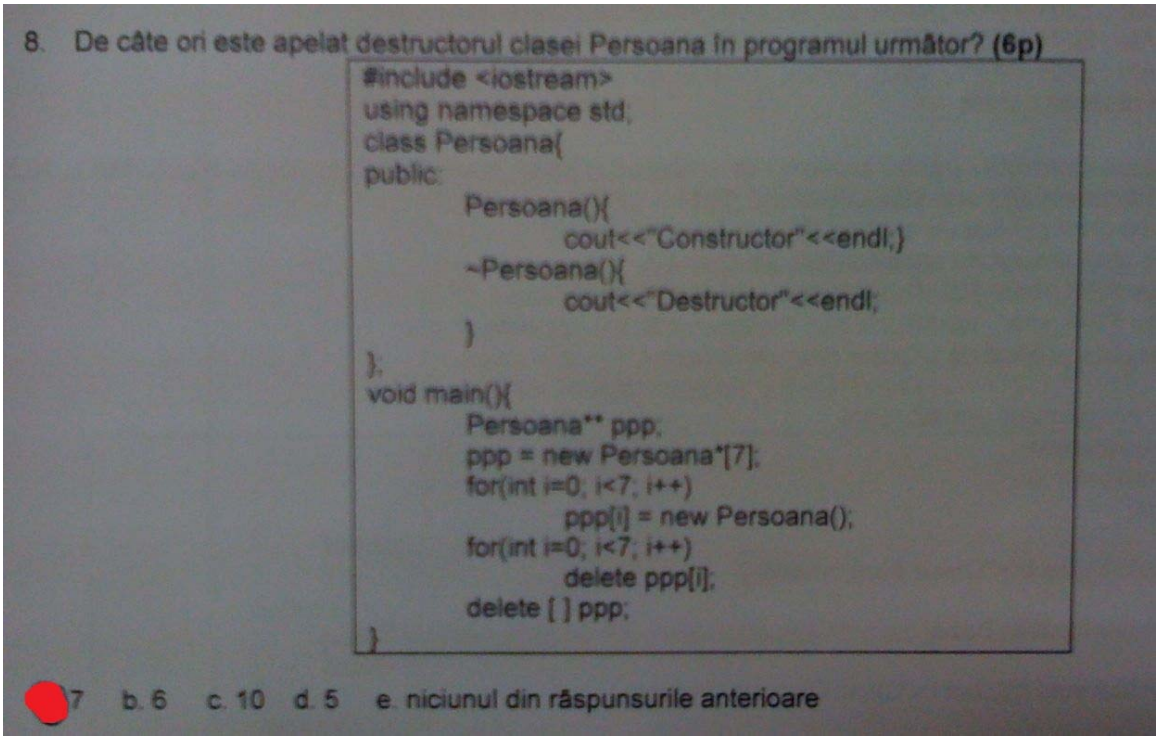
Raspuns E
- o clasa abstracta se defineste folosind cuv cheie abstract
- interfetele nu pot fi create instante ce contin variabile
- clasele ce contine metode abstracte sunt clase abstracte
- clasele abstracte nu pot fi instantiate
- clasele abstracte pot avea variabile care nu sunt statice sau finale.

26.



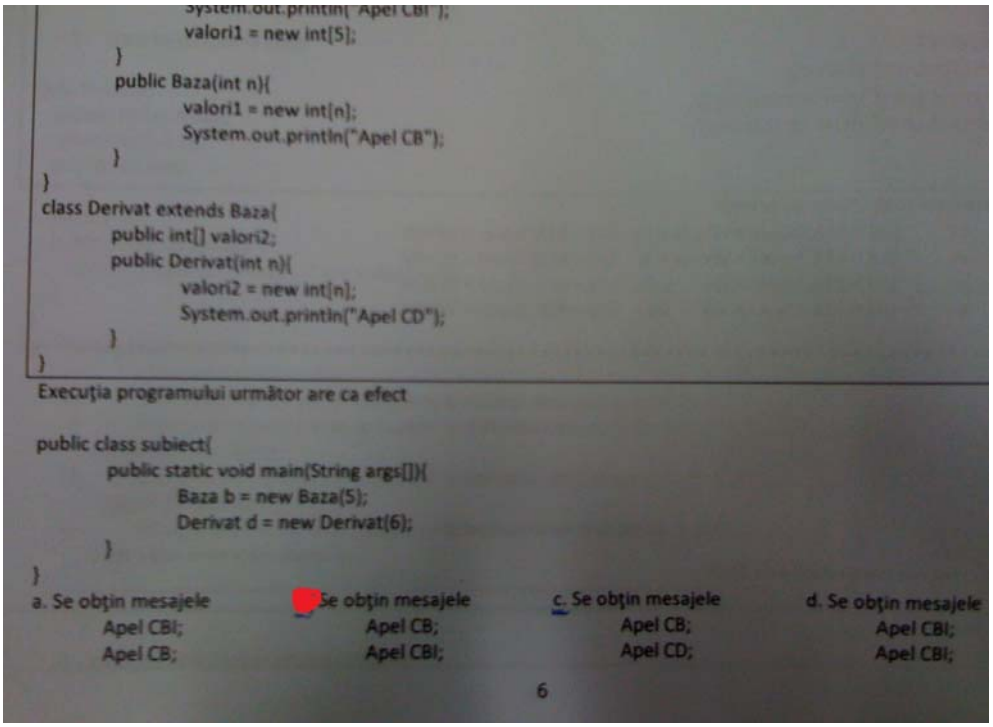
Raspuns B

27.



Raspuns A
Se creaza 7 obiecte in tabloul Persoana si se sterg tot 7 obiecte.
Deci destructorul a fost apelat de 7 ori.

28.



```
            System.out.println("Apel CBI");
            valori1 = new int[5];
        }
        public Baza(int n){
            valori1 = new int[n];
            System.out.println("Apel CB");
        }
}
class Derivat extends Baza{
        public int[] valori2;
        public Derivat(int n){
            valori2 = new int[n];
            System.out.println("Apel CD");
        }
}
```

Execuţia programului următor are ca efect

```
public class subiect{
        public static void main(String args[]){
            Baza b = new Baza(5);
            Derivat d = new Derivat(6);
        }
}
```

a. Se obţin mesajele
Apel CBI;
Apel CB;

b. Se obţin mesajele
Apel CB;
Apel CBI;

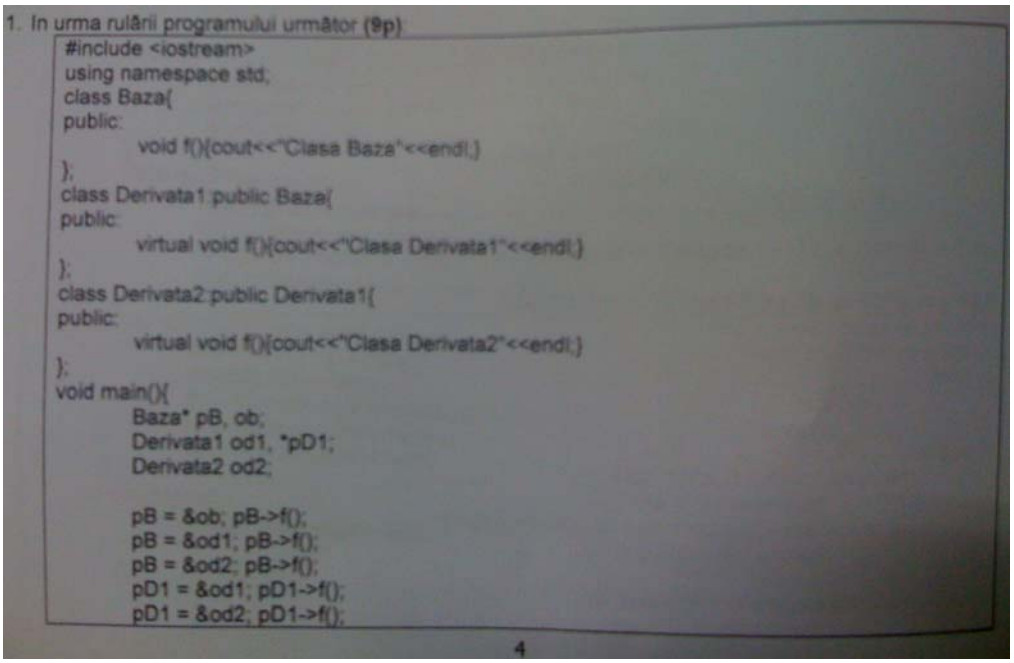
c. Se obţin mesajele
Apel CB;
Apel CD;

d. Se obţin mesajele
Apel CBI;
Apel CBI;

6

Raspuns B:
Clasa de baza contine doi constructori: unul default si unul cu parametru. In metoda main se apeleaza constructorul cu parametru ptr crearea obiectului baza, iar ptr obiectul derivat se apeleaza constructor default + constructorul derivat.
CB CBI CD

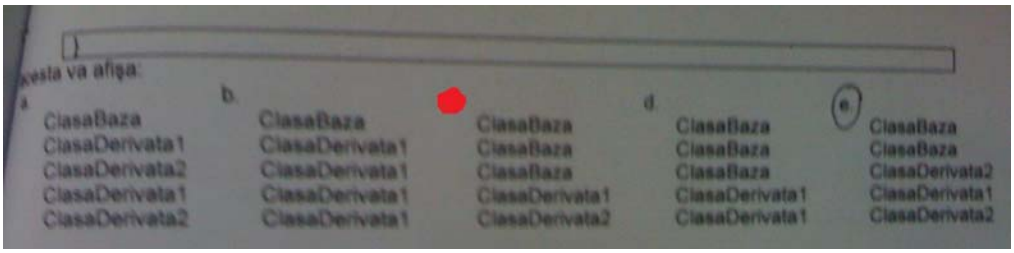
29.



1. In urma rulării programului următor (9p)

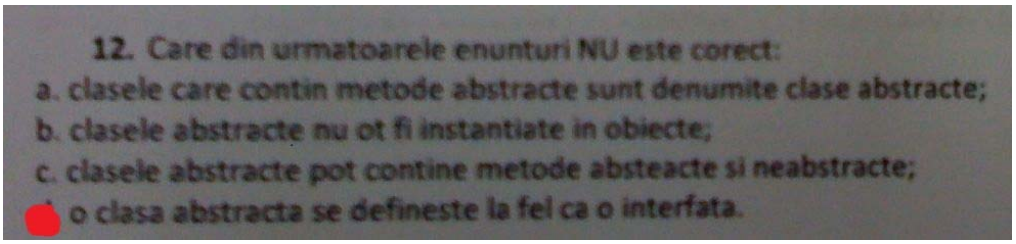
```
#include <iostream>
using namespace std;
class Baza{
public:
        void f(){cout<<"Clasa Baza"<<endl;}
};
class Derivata1:public Baza{
public:
        virtual void f(){cout<<"Clasa Derivata1"<<endl;}
};
class Derivata2:public Derivata1{
public:
        virtual void f(){cout<<"Clasa Derivata2"<<endl;}
};
void main(){
        Baza* pB, ob;
        Derivata1 od1, *pD1;
        Derivata2 od2;

        pB = &ob; pB->f();
        pB = &od1; pB->f();
        pB = &od2; pB->f();
        pD1 = &od1; pD1->f();
        pD1 = &od2; pD1->f();
```

4



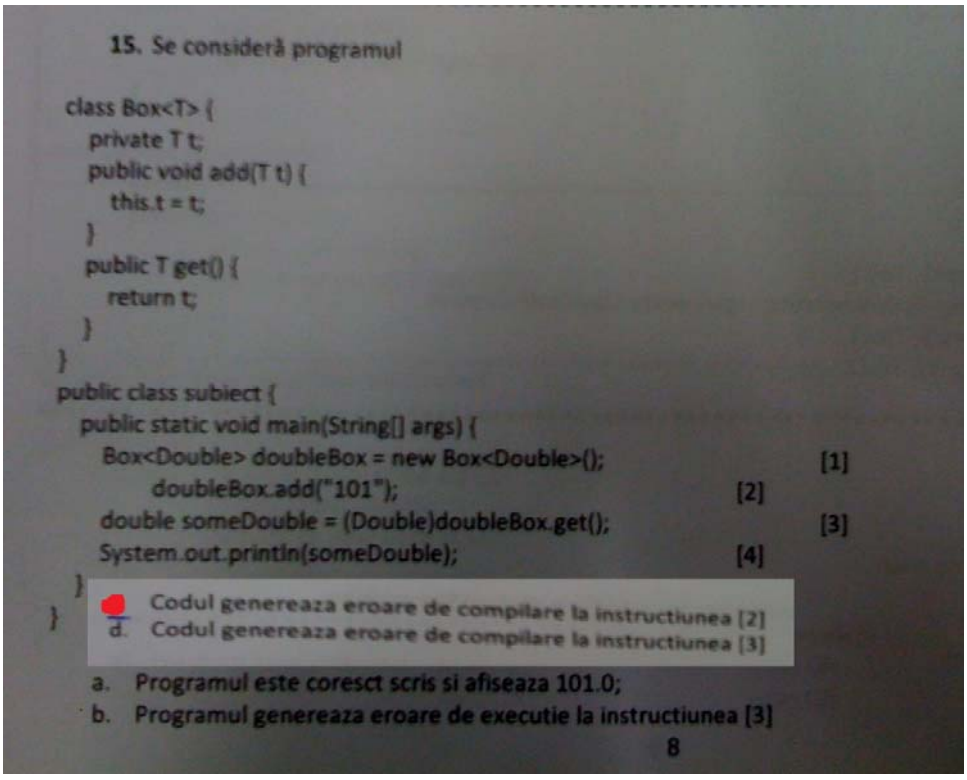
Raspuns C:
Datorita faptului ca clasa de baza nu contine constructor virtual apelarea prin referinta va apela intotdeauna clasa de baza.
In cazul 4 si 5, obiectul derivata1 apeleaza constructorul clasei iar obiectul derivata2 apeleza constructorul clasei derivate2 deoarece derivata1 contine "virtual"

30.



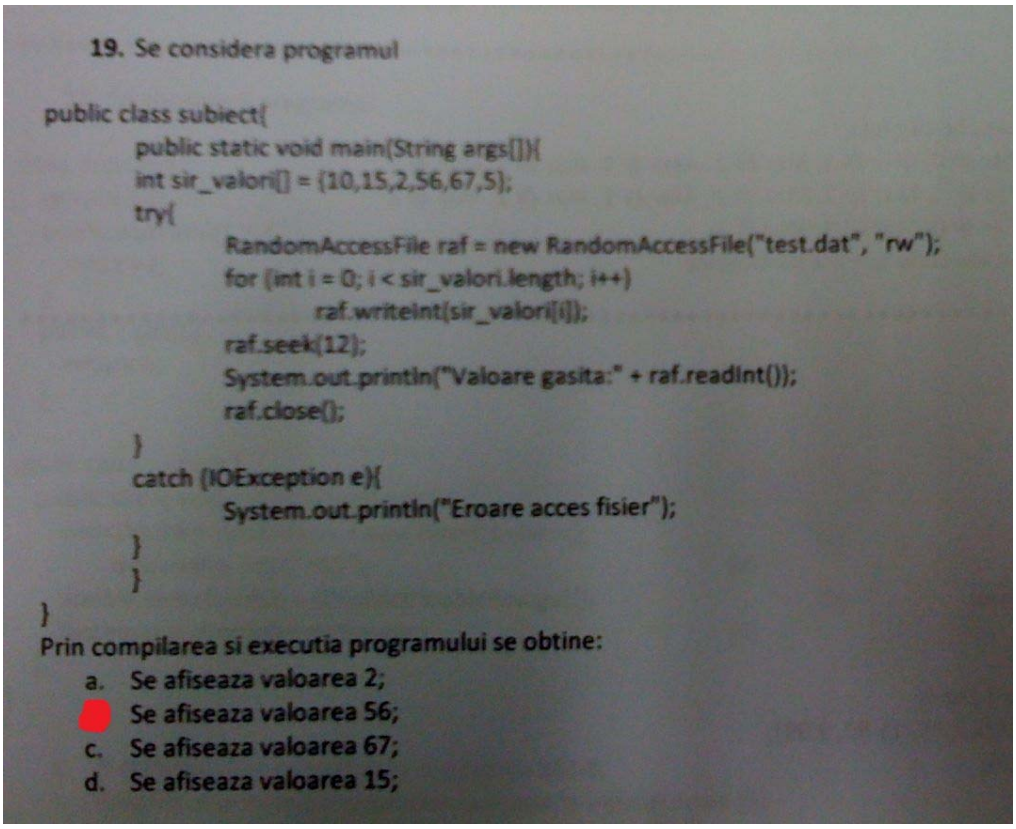
Raspuns :D
Toate celelalte sunt adevarate ...

31.



Raspuns A.
Codul genereaza eroare la linia 2 deoarece valoarea asteptata este de tip double si nu de tip string. Ptr a fi corecta executia trebuie pusa o valoare double - doubleBox.add(101.00);

32.



19. Se considera programul

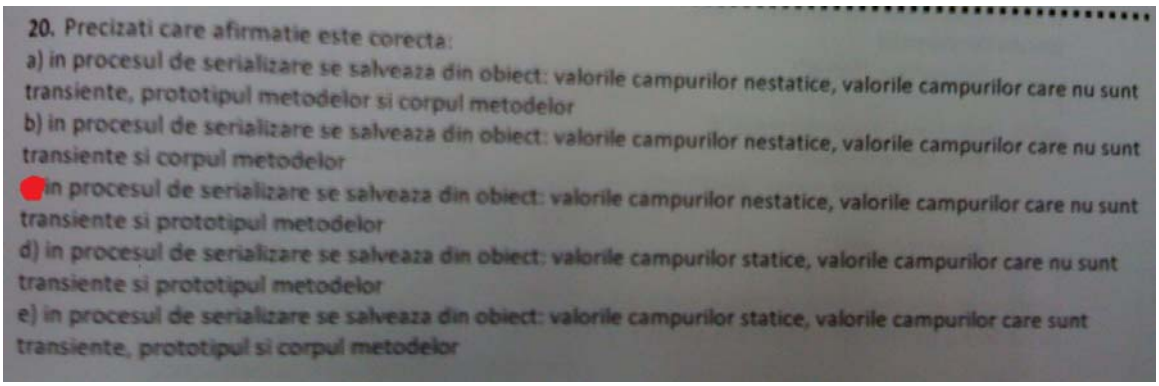
```
public class subiect{
        public static void main(String args[]){
        int sir_valori[] = {10,15,2,56,67,5};
        try{
                RandomAccessFile raf = new RandomAccessFile("test.dat", "rw");
                for (int i = 0; i < sir_valori.length; i++)
                        raf.writeInt(sir_valori[i]);
                raf.seek(12);
                System.out.println("Valoare gasita:" + raf.readInt());
                raf.close();
        }
        catch (IOException e){
                System.out.println("Eroare acces fisier");
        }
        }
}
```

Prin compilarea si executia programului se obtine:

a. Se afiseaza valoarea 2;
b. Se afiseaza valoarea 56;
c. Se afiseaza valoarea 67;
d. Se afiseaza valoarea 15;

Raspuns B
raf.seek in acest caz navigheaza prin sir de valori din 4 in 4.
deci raf.seek[0] ar fi extras valoarea 10.
deci raf.seek[4] ar fi extras valoarea 15, iar raf.seek[12] extrage valoarea 56.

33.



20. Precizati care afirmatie este corecta:
a) in procesul de serializare se salveaza din obiect: valorile campurilor nestatice, valorile campurilor care nu sunt transiente, prototipul metodelor si corpul metodelor
b) in procesul de serializare se salveaza din obiect: valorile campurilor nestatice, valorile campurilor care nu sunt transiente si corpul metodelor
c) in procesul de serializare se salveaza din obiect: valorile campurilor nestatice, valorile campurilor care nu sunt transiente si prototipul metodelor
d) in procesul de serializare se salveaza din obiect: valorile campurilor statice, valorile campurilor care nu sunt transiente si prototipul metodelor
e) in procesul de serializare se salveaza din obiect: valorile campurilor statice, valorile campurilor care sunt transiente, prototipul si corpul metodelor

Raspuns C

34.

21. Se considera programul

```
class exceptieAdunare extends Exception{
        public exceptieAdunare(String mesaj) { super(mesaj); }
}
public class subiect{
        public static int aduna(int a, int b) throws exceptieAdunare{
                if (b == 0) throw (new exceptieAdunare("Eroare adunare !"));
                else
                        return a + b;
        }
        public static void main(String args[]){
                int vb1 = 100;
                int vb2 = 0;
                int rez;
                try{
                        rez = aduna(vb1, vb2);
                        rez = vb1 / vb2;
                        rez = 100;
                }
                catch (ArithmeticException e){
                        rez = 0;
                }
                catch (Exception e){
                        rez = +1;
                }
                System.out.println("Valoarea rezultatului este "+rez);
        }
}
```

Programul afiseaza:

a. Valoarea rezultatului este 0;

b. Valoarea rezultatului este 100;

Valoarea rezultatului este 1;

d. Programul genereaza exceptii care nu sunt captate si tratate

e. Valoarea rezultatului este NaN

Raspuns C

vb2 este egal cu 0, deci vom avea exceptie. Avand in vedere ca avem o exceptie definita din Exception, rezultatul va fi : rez + 1, deci valoarea finala va fi 1.

35.

Question 1 of 40 — 2.0 Points

Given:

```
...
        try {
             srandom = SecureRandom.getInstance("SHA1PRNG","SUN");
             srandom.setSeed(seed);

             keyPairGenerator = KeyPairGenerator.getInstance("RSA");
             keyPairGenerator.initialize(1024, srandom);

             rsaKeyPair = keyPairGenerator.generateKeyPair();
             rsa = Signature.getInstance("SHA1withRSA");
             rsa.initSign(rsaKeyPair.getPrivate());
...
```

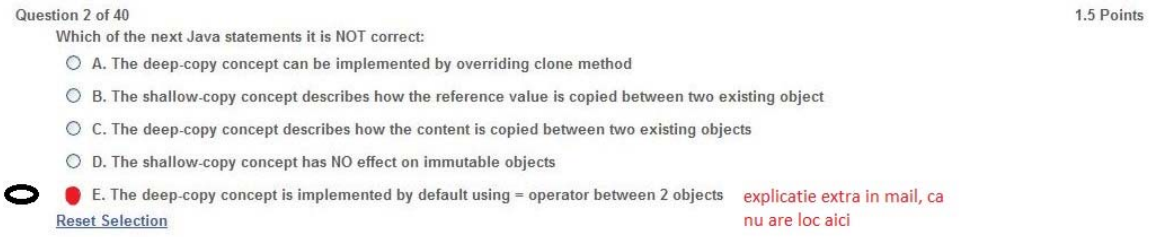
select the correct description

- A. generates a public and private key for DSA over SHA-1 hash — nu e asta pentru ca nu foloseste DSA, ci RSA
- B. it is initialized an "srandom" object using the Bouncy Castle provider — nu foloseste Bouncy Castle (care s-ar fi prescurtat "BC" in loc de "SUN"), ci providerul default de la Sun
- C. generates a public and private key for RSA over MD5 hash — nu se fol MD5, ci SHA1
- D. none of these descriptions is correct
- E. it is initialized an "srandom" object using the default Java Sun provider — este corect, se foloseste providerul default de la Sun de librarii criptografice

Reset Selection

Raspuns E

36.



Notiunea de "deep copy" (care este opusul lui "shallow copy") intre 2 obiecte atat in Java, cat si in orice alt limbaj de programare:

- am clasele:

```
Public class Adresa{

        String strada;

        String oras;

        String tara

}

Public class Student{

        String nume;

        Int varsta;

        Adresa adresa;// care e la randul lui un obiect cu strada, oras, tara ca si
campuri

}
```

- Am un student: Student s1 = ... (ceva)

care are campuri simple nume, prenume, dar si un camp compus, sa spunem Adresa ();

- daca eu vreau sa contruiesc un alt student s2 prin copierea campurilor din primul: este ok sa spun ca s2.nume = s1.nume, dar nu este ok sa spun s2.adresa = s1.adresa pentru ca astfel am creat o copie shalow; cu alte cuvinte, in momentul asta, ambele campuri de adresa pentru cei 2 studenti bat in acelasi obiect, iar daca o voi modifica pe oricare dintre ele, se va modifica si celalalt, lucru pe care nu mi-l doresc; ca sa creez o copie „deep” ar fi trebuit sa fac astfel:

```
S2.adresa = new Adresa();//am alocat spatiu pentru un nou obiect

S2.adresa.strada = s1.adresa.strada;//copiez valorile obiectului, nu obiectul in sine
```

```
S2.adresa.oras = s1.adresa.oras;

S2.adresa.tara = s1.adresa.tara;
```

La campurile nume si varsta nu este valabil pentru ca:

- Varsta este int (tip de data primitiva, nu este un obiect, iar pentru acestea este ok sa folosesc operatorul =);
- Nume este String=> este obiect immutable, iar pentru astea shallow copy nu are efect

(obiect „immutable” = nu-si poate modifica valoarea, adica:

Daca am

```
String s1 = „Test”;

s1 = s1 + „asd”; // in momentul asta s1 devine „Testasd”, dar nu pentru ca s-a modificat vechiul obiect, ci pentru ca s-a creat unul nou, iar cel vechi urmeaza sa fie luat de Garbage Collector; toate String-urile sunt immutable)
```

37.

Question 3 of 40 — 2.0 Points

Given:

```
public class Main {

    public static void main(String[] args) {
        double[] frequency = {10.5, 3.45, 15.99, 3.99};
        int[] values = {5, 8, 11, 29};
        try {
            DataOutputStream out = new DataOutputStream(new BufferedOutputStream(new FileOutputStream("test.txt")));
            for (int i = 0; i < frequency.length; i++) {
                out.writeDouble(frequency[i]);
                out.writeInt(values[i]);
            }
            out.flush();
            DataInputStream in = new DataInputStream(new BufferedInputStream(new FileInputStream("test.txt")));
            try {
                while (true) {
                    int val = in.readInt();
                    double frecv = in.readDouble();

                    System.out.format("Value %d - %.2f frequency ", val, frecv);
                }
            } catch (EOFException e) {
                //aici iese cand s-a terminat fisierul
            }
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }
}
```

Este corect B, l-am si testat pe calculator anul trecut. Ideea e ca el acolo scrie in fisier intai un double (cu virgula), iar apoi un intreg, dupa care, cand le citeste, le ia in ordine inversa. Astfel, el le citeste prost si ii dau valorile alea aberante.
! Ai grija, ca imi aduc aminte ca anul trecut, cand am dat noi examenul, a existat aceasta intrebare, doar ca modificata si se citeau bine; adica se scriau: intreg, double si se citeau in aceeasi ordine, tot intreg, double; iar in cazul ala nu mai dadeau valori aberante, ci exact valorile initiale;

what is printed ?

- A. the example does't print something because the text file is empty
- B. Value 1076166656 - 0,00 frequency Value 1074502041 - -0,00 frequency Value 1076886241 - 19991590273549737000000000000000000000,00 frequency Value 1074785157 - 0,00 frequency
- C. the example generates runtime exceptions
- D. the example generates compiler errors
- E. Value 5 - 10,50 frequency Value 8 - 3,45 frequency Value 11 - 15,99 frequency Value 29 - 3,99 frequency

Reset Selection

Raspuns B
Similar cu intrebarea nr. 12, doar ca sunt inversate tipurile de date la citire compativ cu cele la scriere. Si astfel se obtin valori ciudate.

38.

Question 4 of 40 — 3.0 Points

Given:

```
public class Main {
    public static void main(String[] args) {
        String s1 = "abc";
        String s2 = s1;
        s1 += "d";
        System.out.println(s1 + " " + s2 + " "+ (s1 == s2));

        StringBuffer sb1 = new StringBuffer("abc");
        StringBuffer sb2 = sb1;
        sb1.append("d");
        System.out.println(sb1 + " " + sb2 + " "+ (sb1 == sb2));
    }
}
```

which is true ?

- A. abcd abc false abcd abcd false
- B. abc abc false abcd abcd true
- C. abcd abc false abcd abcd true
- D. abc abc true abcd abcd true
- E. abcd abc true abcd abcd true
- F. abcd abc false abcd abc true

Reset Selection

Ai dreptate, este corect C pentru ca:
- atunci cand s-a facut: s1+="d" (care inseamna s1=s1+"d") obiectul catre care batea s1 nu s-a modificat, ci s-a creat unul nou, cu valoarea "abcd" (string este ummutable). Deci s1 si s2 nu mai sunt egale, s2 ramanand nemodificat.
In cazul folosirii clasei StringBuffer nu mai este valablil acest lucru. Cand am scris sb2 = sb1 insemana ca ambele referinte bat in acelasi obiect, iar daca il modific pe unul, se modifica si valoarea celuilalt. In plus,cele 2 obiecte sunt egale, adica se afla la aceeasi adresa, astfel ca se returneaza true la evaluarea conditiei sb1==sb2.
!!!OBS:
Daca aveam StringBuffer sb2 = new StringBuffer(sb1) - creez un NOU obiect dandu-i acelasi continut ca primului;
Astfel ca daca il modificam pe unul dintre ele, celalalt nu se modifica. Mai mult, la evaluarea expresiei sb2==sb1 in acest caz, as avea false. Cele 2, desi au acelasi continut (au acelasi String in ele), nu reprezinta acelasi obiect. Poti sa le consideri obiecte identice ca si aspect, dar nu sunt ACELASI obiect. In cazul obiectelor, operatorul == compara referintele/adresele, nu compara valorile (compara valorile doar in cazul datelor promitive: string, int, floar, double, etc)

Raspuns C

39.

Question 6 of 40 — 4.0 Points

In following program:

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

class Person {
 int age;
 char * name;
public:
 Person(int v=0, char* n="Anybody") : age(v) {
  this->name = new char [strlen(n)+1];
  strcpy(this->name, n);
  cout<<"Constructor"<<endl;
 }

 Person(Person & p) {
  this->age = p.age;
  this->name = new char [strlen(p.name)+1];
  strcpy(this->name, p.name);
  cout<<"Copy constructor "<<endl;
 }

 void operator = (Person & p) {
  this->age = p.age;
  delete [ ] this->name;
  this->name = new char [strlen(p.name)+1];
  strcpy (this->name, p.name);
  cout<<"Operator ="<<endl;
 }

 ~Person ( ) {
  cout<<"Destructor"<<endl;
 }
};

void main ( ) {
 Person p1, p2(20, "Smith"); - se apeleaza contructorul de 2 ori, prima data cu valorile implicite, v=0 si n="nobody", iar a doua oara cu parametri
 Person p3 = p1;  - se apeleaza contructorul de copiere, si se creeeaza un nou obiect avand valorile copiate din p1
 p3 = p2; - se apeleaza operatorul =, iar valorile atributelor lui p2 sunt copiate in p3
 Person p4 = p1; - se apeleaza din nou contructorul de copiere, la fel ca mai sus
 p1 = p2; - se apeleaza din nou operatorul egal
} - se apeleaza destructorul pentru toate cele 4 obiecte create, p1, p2, p3, p4; destructorul elibereaza, in general toate resursele alocate in timpul progr
```

what are the number of calls?

- A. constructor – 2 times, copy constructor – 2 times, operator = 2 times, destructor – 4 times
- B. constructor – 4 times, copy constructor – 1 time, destructor – 4 times
- C. constructor – 2 times, copy constructor – 2 times, operator = 1 time, destructor – 4 times

Raspuns A:
Similara cu problemele 1 si 7.

40.

Question 7 of 40 2.0 Points

Using OpenSSL library to verify a RSA electronic signature, which of the following OpenSSL functions is not use within an application for electronic signature verification

- A. MD5_Update
- B. PEM_read_RSAPublicKey
- C. RSA_public_decrypt
- D. RSA_private_encrypt
- E. MD5_Init

Reset Selection

Raspuns :D - vezi cod C++

41.

Question 8 of 40 3.0 Points

In the following program:

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

class Car{
private:
 int prodYear;
 char * color;
public:
 Car ( ) {
  cout<<"default constructor (no parameters)"<<endl;
 }

 Car (int year = 0, char * col = "") {
             cout<<"constructor with parameters having default values"<<endl;
  this->prodYear = year;
  this->color = new char[strlen(col)+1];
  strcpy(this->color, col);
 }

 ~Car ( ) {
  cout<<"destructor"<<endl;
  delete[ ] this->color;
 }
};

void main ( ) {
 Car m1;               //1
 Car m2(2000,"White"); //2
 Car m3(1000);         //3
 cout<<m1.prodYear<<" "<<m1.color<<endl;  //4
 cout<<m2. prodYear <<" "<<m2.color<<endl;//5
 cout<<m3. prodYear <<" "<<m3.color<<endl;//6
}
```

1. da eroare pentru ca nu stie pe care dintre contructori sa-l apeleze: pe cel default sau pe cel cu valori implicite; oricare dintre ele ar fi corect, dar findca exista ambele, aplicatia crapa;
4, 5 si 6 dau eraore pentru ca incearca sa acceseze campul color (care este privat) al celor 3 obiecte create;

what are the code lines in main( ) function generating compiling errors?

- A. 3 + 4 + 5 + 6
- B. 4 + 5 + 6
- C. 1 + 4 + 5 + 6
- D. 1 + 3 + 4 + 5 + 6
- E. 1 + 3

Reset Selection

Raspuns C:
Similara cu problema 2.

42.

Question 9 of 40 2.0 Points

Given:

```
class Something{
    int[] someValues = {10,20,30};
}
public class Main {
    public static void main(String[] args) {
1       Something [] some = new Something[3];
2       some[0] = new Something();
3       Something aThing = new Something();
4       some[1] = aThing;
5       aThing = null;
6       some[1] = null;
        //other things
    }
}
```

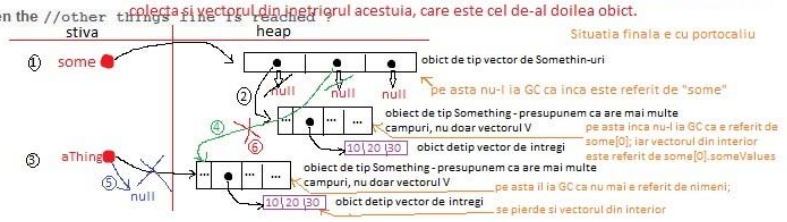
randul asta arata ca un obict de tip Something are la randul sau(in interiorul lui) un alt obiect, de tip vector

1. Aici se creeaza un obiect, denumit some, care reprezinta un vector(container) de obiecte Something (pana in punctul asta, niciun obiect de tipul SOmething NU a fost create, ci doar s-a creat spatiu pentru acestea)
2. Se creeaza un obiectde tipul Something, a carui adresa se pune pe prima pozitie a vestorului
3. Se creeaza un nou obiect, denumit aThing, de tipul Something
4. NU se mai creeaza niciun obiect; singurul lucru care se face aici este: adresa obiectului aThing este copiata pe pozitia a doua a vectorului.
5. referinta aThing nu mai pointeaza catre obictul in sine ci a fost facuta null;in momentul asta, some[1] inca pointeaza catre obiect, deci referinta catre el nu s-a pierdut
6. referinta some[1] nu mai pointeaza nici ea catre obiect, iar in mom. asta nu mai avem nicio refernita care sa duca in obiect, asa ca GC va veni sa-l colecteze. mai mult decat atat, GC va colecta si vectorul din interiorul acestuia, care este cel de-al doilea obict.

how many objects are eligible for GC when the //other things line is reached ?

- A. 2
- B. 3
- C. 5
- D. 4
- E. 1

Reset Selection



Raspuns A - posibil si E ???

43.

Question 10 of 40 2.0 Points

Given

```
class BankAccount{
    Integer amount = 200;
    BankAccount doSomething(BankAccount ba){
        ba = null;
        return ba;
    }
}
public class Main {
    public static void main(String[] args) {
        BankAccount ba1 = new BankAccount();
        BankAccount ba2 = new BankAccount();
        BankAccount ba3 = ba1.doSomething(ba2);
        ba1 = null;
        //other things
    }
}
```

se creeaza primul obiect
se creeaza si cel de-al doilea
doSomething modifica referinta ba2, facand-o null; aici nu se creeaza niciun obiect, este ca si cand as fi spus BankAccount ba3 = null;
se modifica si referinta ba1

Asa ca, la final, GC are de colectat 2 obicte, create in primele 2 randuri si care nu mai sunt referite acum de nimeni

when //other things is reached, how many objects are eligible for GC ?

- A. runtime exception
- B. 1
- C. compilation errors
- D. 3
- E. 2
- F. it is not possible to determine

Reset Selection

Raspuns E

44.

Question 11 of 40 2.0 Points

For the next command line:

```
keytool.exe -genkey -keyalg RSA -alias ISMCert1 -keypass parolaism1 -storepass parolaks -keystore keystoreISM1.ks -dname
"cn=IT&C Security Master, ou=IT&C Software Development, o=RO"
```

select the correct description

- A. Generates a public and private key store named "keystoreISM1.ks" that has the password "parolaks"
- B. Generates a public and private key store named "ISMCert1" that has the password "parolaks"
- C. Generates a RSA public and private key named "keystoreISM1.ks" that has the password "parolaks"
- D. Generates a RSA public and private key named "ISMCert1" that has the password "parolaks"
- E. Generates a public and private key store named "keystoreISM1.ks" that has the password "parolaism1"

Reset Selection

Raspuns A

45.

Question 12 of 40 — 2.0 Points

Which of the below assigning operations is incorrect, *ch* being of type *char*:

- A. ch = '\0';
- B. ch = '\x30';
- C. ch = 'x';
- D. ch = A; A is a variable of type bool
- E. ch = "T";

Nu am o explicatie prea logica pentru asta, doar stiu ca l-am incercat pe calculator si asta e corect in C/C++; ideea e ca tip de date "char" inseamna ca accepta doar un caracter, iar ghilimelele indica un sir de caractere, chit ca tu ii dai acolo unul singur.

Reset Selection

Raspuns E

46.

Question 13 of 40 — 2.0 Points

Given:

```
class Something{
    int[] someValues = {10,20,30};
}
public class Main {
    public static void main(String[] args) {
        Something [] some = new Something[3];
        some[0] = new Something();
        Something aThing = new Something();
        some[1] = aThing;
        aThing = null;
        some[1] = null;
        //other things
    }
}
```

how many objects are created ?

- A. 5
- B. 3
- C. 2
- D. 4
- E. 1

e acelasi program ca mai inainte, asa ca logica e aceeasi; daca numeri din desenul ala facut de mine, o sa vezi ca sunt 5 obicte create in total pe parcursul acelui program

! Obs: pentru a numara obiectele, trebuie sa te uiti doar in partea din dreapta a desenului, in zona denumita heap.

Reset Selection

Raspuns A - Similara cu problema 42

47.

Question 14 of 40 — 2.0 Points

The next Java example:

```
class MyException extends Exception{
        public MyException(String Message) { super(Message); }
    }
public class Main {
    public static void main(String[] args) {
        // TODO code application logic here
        int a = 0, b = 0, c = 0;
        try{
                a = 10; b = 20; c = 30;
                if(b == 20) throw new MyException("Test");
                c = 40;
        }
        catch (MyException e){
         System.out.println(e.getMessage());
            a = 22;
        }
        catch (Exception e){
            System.out.println(e.getMessage());
            b = 33;
        }
        finally{
                a = 99;
        }
        System.out.println("a = "+a+" and b = "+b+" and c ="+c);
    }
}
```

de aici sare direct pe myexception, restul pana acolo nu se mai executa

in acest catch nu intra, pentru ca a intrat in primul

blocul finally se executa intotdeauna, si in caz de eroare si in caz contrar

generates the result:

- A. a = 99, b = 20, c = 40;
- B. a = 99, b = 33, c = 30;
- C. a = 99, b = 20, c = 30;
- D. a = 99, b = 33, c = 40;
- E. a = 10, b = 20, c = 30;

Reset Selection

Raspuns C

48.

Question 15 of 40 — 2.0 Points

The next class:

```
class Student{
    private String nume;
    private int[] note;
    private int cod;

    @Override //annotation
    public String ToString()
    {
        return "Studentul "+nume+" are codul "+cod;
    }
}
```

- A. generates a compilation error because the ToString() method is accessing private fields — orice metoda poate sa acceseze orice camp al clasei, fie el privat, pentru ca se afla in interior;
- B. generates a compilation error because the ToString() does NOT override a method inherited from Object — ba fix asta face, suprascrie metoda din Object
- C. generates a compilation error because the @Override annotation is used only for classes and NOT for methods — gresit, e doar pt metode
- D. generates a compilation error because the @Override annotations force us to extend explicitly the Object class — gresit
- **E. The class is written correctly**

Reset Selection

Raspuns E

49.

Question 16 of 40 — 2.0 Points

The next Java example

```
class Base{
    public int[] valori1;
    public Base(){
        System.out.println("DBC call");
        valori1 = new int[5];
    }
    public Base(int n){
        valori1 = new int[n];
        System.out.println("BC call");
    }
}
class Subclass extends Base{
    public int[] valori2;
    public Subclass(int n){
        valori2 = new int[n];
        System.out.println("SC call");
    }
}
public class Main {
    public static void main(String[] args) {
        Base b = new Base(5); // apel de constructor din Base cu 1 parametru (BC)
        Subclass d = new Subclass(6); // apel de constructor cu 1 parametru din Subclass, care face prima data un apel de constructor default din Base (si scrie "DBD call"), iar abia apoi scrie si "SC call"
    }
}
```

prints

- **A. BC call
DBC call
SC call**
- B. DBC call
BC call
SC call
- C. BC call
BC call
SC call
- D. BC call
SC call
- E. BC call
SC call
DBC call

Reset Selection

Raspuns A - similara cu problema 28 , doar ca textul afisat este diferit.

50.

Question 17 of 40 — 2.0 Points

Given:

```
class Container{
    private String name;
    private static Container instance = null;
    private Container(){
        this.name = "Nothing";
    }
    public static Container getInstance(){
        if (instance == null){
            instance = new Container();
        }
        return instance;
    }
    public void setName(String x) { this.name = x; }
    public String getName() { return this.name; }
}

public class Main{
    public static void main(String[] args){
        Container s1 = Container.getInstance();
        Container s2 = Container.getInstance();
        s1.setName("Container 1");
        s2.setName("Container 2");
        System.out.println("s1=" + s1.getName() + " s2=" + s2.getName());
    }
}
```

Acesta este un exemplu clasic de clasa Singleton:
- are constructorul "private" => nu se poate contrui niciun obiect din afara acestei clase
- are o metodat statica (getInstance()) prin care se creeeaza un obict, daca nu a mai fost niciunul creat pana atunci, iar daca deja exista un obict, acesta se refoloseste.

Astfel ca getInstance() intoarce acelasi obict pentru ambele referinte s1 si s2. Cand o modific pe una, se modifica si cealalta.

what is printed ?

- A. s1=Container 1 s2=Container 1
- B. s1=Container 2 s2=Container 2
- C. s1=Container 2 s2=Container 1
- D. compiler errors because the constructor is private
- E. s1=Nothing s2=Nothing
- F. s1=Container 1 s2=Container 2

Reset Selection

Raspuns B

51.

Question 18 of 40 — 3.0 Points

Given:

```
class Test {
    public static String s = "*";
    void s1()
    {
        try
        {
            s2();
        }
        catch (Exception e)
        {
            s += "c";
        }
    }
    void s2() throws Exception {
        s3();
        s+= "2";
        s3();
        s+="2b";
    }
    void s3() throws Exception{
        throw new Exception();
    }
}
public class Main {
    public static void main(String[] args) {
        new Test().s1();
        System.out.println(Test.s);
    }
}
```

In blocul try apeleaza metoda s2(); aceasta apeleaza metoda s3; metodat s3 arunca o exceptie; ne intorcem in metoda s2(), dar s-a aruncat o exceptie; cautam sa vedem daca metoda dispune de un mecanism try-catch de prindere a exceptiilor => NU are => arunca exceptia mai departe, in sus (si nu se mai executa nimic dupa acel apel a3(); ne intorcem in metoda s1, dar s-a aruncat o exceptie; cautam din nou sa vedem daca metoda dispune de un mecanism try-catch => Da => intra pe catch si adauga doar caracterul c).

what is the result ?

- A. *2c
- B. *c2
- C. *2c2b
- D. *c
- E. *c22b
- F. *

Reset Selection

Raspuns D

52.

Question 19 of 40 — 2.0 Points

It considers the function call `RSA_private_decrypt(RSA_size(A), B, C, D, RSA_PKCS1_PADDING);` Which of the following situations correctly places the private key in the call?

- A. The private key is stored in parameter B
- B. The private key is stored in parameter D
- C. The private key is stored in parameter A
- D. Function `RSA_private_decrypt` is not define in OpenSSL library
- E. The private key has not to be specified in parameter list

Reset Selection

Raspuns : D - RSA_private_decrypt(RSA_size(apub), e_data, last_data, apriv, RSA_PKCS1_PADDING);

53.

Question 20 of 40 — 3.0 Points

Please, specify what the following program displays:

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

class Car {
 int prodYear;
 char * color;
public:
 Car (int year = 0, char* col = "") {
  this->prodYear = year;
  this->color = new char [strlen(col)+1];
  strcpy(this->color, col);
 }

 Car & operator= (Car & m){
  this->prodYear = m.prodYear;
  delete [ ] this->color;
  this->color = new char [strlen(m.color)+1];
  strcpy(this->color, m.color);
  return (*this);
 }

 int getProdYear ( ) { return this->prodYear; }

 void setProdYear (int year) { this->prodYear = year; }

 char * getColor ( ) { return this->color; }

 void setColor (char * c) {
  delete [ ] this->color;
  this->color = new char [strlen(c)+1];
  strcpy(this->color, c);
 }
};

void main ( ) {
 Car c1(2000,"White");
 Car c2(2001,"Black");
 Car c3 = c2;
 Car c4(2003, "Red");
 c3.setColor("Green");
 c4 = c1;
 cout<<c1.getProdYear ( )<<" "<<c1.getColor ( )<<" ; ";
 cout<<c2.getProdYear ( )<<" "<<c2.getColor ( )<<" ; ";
 cout<<c3.getProdYear ( )<<" "<<c3.getColor ( )<<" ; ";
 cout<<c4.getProdYear ( )<<" "<<c4.getColor ( )<<" ; ";
}
void main ( ) {
 Car c1(2000,"White");
 Car c2(2001,"Black");
 Car c3 = c2;  se apeleaza constructorul de copiere;  dar se creaza o referinta la obiectul c2 , deci modificarea lui c3 va modifica valorile lui c2
 Car c4(2003, "Red");
 c3.setColor("Green");  setamculoarea lui c3, dar c2   se modifica
 c4 = c1;   se copiaza obiectul c1 in c4
 cout<<c1.getProdYear ( )<<" "<<c1.getColor ( )<<" ; ";
 cout<<c2.getProdYear ( )<<" "<<c2.getColor ( )<<" ; ";
 cout<<c3.getProdYear ( )<<" "<<c3.getColor ( )<<" ; ";
 cout<<c4.getProdYear ( )<<" "<<c4.getColor ( )<<" ; ";
}
```

- A. 2000 White; 2001 Green ; 2001 Green ; 2000 White ;
- B. 2000 White ; 2001 Black ; 2001 Black ; 2000 White ;
- C. 2000 White ; 2001 Black ; 2001 Green ; 2000 White ;
- D. none of the previous responses
- E. 2000 White ; 2001 Black ; 2001 Green ; 2000 Red ;

Raspuns : C

54.

Question 21 of 40 2.0 Points

For the next statement:

```
javax.crypto.Cipher cipher = javax.crypto.Cipher.getInstance("DES/ECB/KEY","BC");
```

what is the correct description

- A. It is created a DES cipher, used in ECB mode, with a given private key,using Bouncy Castle provider
- B. the instruction generates a NoSuchAlgorithmException at runtime
- C. It is created a AES cipher, used in ECB mode, with a given private key,using Bouncy Castle provider
- D. the instruction generates a NoSuchPaddingException at runtime (PARCA) metoda aia vrea algoritm/mod/padding iar "key" nu era un padding cunoscut
- E. It is created a DES cipher, used in CBC mode, with a given private key, using Sun provider

Reset Selection

Raspuns : D - Cipher.getInstance("AES/CBC/PKCS5Padding", provider);

55.

Question 22 of 40 2.0 Points

After running the program:

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

void f(int x[ ], int len) {
 for(int i=0; i<len; i++)
  if(x[i]%2 == 0)
   cout<<x[i]<<" ";
}

void f(int len, int* x) {
 for(int i=0; i<len; i++)
  if(x[i]%2 != 0)
   cout<<x[i]<<" ";
}

void main() {
 int* v = new int [10];
 for(int i=0; i<10; i++)
  v[i] = i;
 f(10, vect);
}
```

- apelul f(10,vect) intra pe aceasta a doua metoda, pentru ca este cea care accepta ca parametri intai un intreg si apoi un vector (care aici este reprezentata printr-un pointer(*) = adresa de inceput a masivului);
- vectorul care este dat ca parametru functiei (vect) contine toate numerele de la 0 la 9;
- functia parcurge vectorul si verifica daca restul impartirii numarului curent din vector este diferit de 0, adica este numar impar (x[i]%2 reprezinta restul impartirii lui x[i] la 2) si le afiseaza doar pe cele impare.
- astfel ca la final vom avea 1 3 5 7 9

- A. the values 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 are displayed;
- B. the values 0 2 4 6 8 are displayed;
- C. ambiguity error is reported at compiling time;
- D. the values 1 3 5 7 9 are displayed;
- E. ambiguity error is reported at run-time;

Reset Selection

Raspuns :

56.

Question 23 of 40 2.0 Points

For this class:

```
class Automobil{
    private String name;
    public int cc;
    public int id = 1;
    private static int noAutos = 0;
    public final String engine = "gasoline";

     public Automobil(){
        noAutos++;
        id = noAutos;
        name = "Nothing";
        cc = 1400;
        engine = "diesel";
    }
}
```

valoarea unui camp definit ca fiind final nu poate fi modificata, odata ce a fost definita;
- exista 2 modalitati de a asigna o valoare unui camp final:
1. direct la declarare, asa cum este facut in cazul nostru
2. in constructor, in cazul in care NU a fost definita valoarea la declarare

- A. the constructor is ok and the class is defined without errors
- B. the constructor generates compiler errors because it changes the engine field value
- C. the constructor generates compiler errors because it changes the id field value
- D. the constructor generates compiler errors because it changes the noAutos static field and engine field values
- E. the constructor generates compiler errors because it changes the noAutos static field value

Reset Selection

Rasp: B

57.

Question 24 of 40 — 2.0 Points

The next Java program that uses Integer objects:

```
public static void Modify(Integer a, Integer b, Integer c){
         b = 20;
         c = 30;
         a = c;
 }
 public static void main(String[] args) {
        Integer a = 10, b = 2, c = 3;
        Modify(a, b, c);
        System.out.println("a = "+a+" and b = "+b+" and c ="+c);

}
```

- faptul ca eu ii pasez niste parametri de intrare metodei Modify, inseamna ca pe stiva metodei s-au copiat valorile celor 3 obiecte. Ele se interschimba in cadrul functiei, dar cand metoda isi incheie executia, stiva acesteia se sterge, iar valorile interschimbate se sterg si ele; => inapoi in metoda main voi avea vechile valori ale celor 3 obiecte, a,b si c;

generates the results:

- (●) A. a = 10 and b = 2 and c = 3
- ( ) B. the source code has compiler errors because the variables a,b,c are modified inside the method.
- ( ) C. a = 10 and b = 20 and c = 3;
- ( ) D. a = 10 and b = 20 and c = 30;
- ( ) E. a = 30 and b = 20 and c = 30;

Reset Selection

Raspuns A: Este la fel ca problema 24.

58.

Question 25 of 40 — 2.0 Points

What are the displayed values after running the following program?

```
#include <stdio.h>

void main(){
 char * pv;
 char c[ ] = { 'a','b', 'c' };
           pv = c;
 printf("%d %d", sizeof(pv), sizeof(*pv));
}
```

- ( ) A. a compiling error is reported because the loading of the pointer is incorrect; the correct version is *pv =&c*
- ( ) B. 2 12
- (●) C. 4 1
- ( ) D. 2 4
- ( ) E. 4 4

Reset Selection

Asta pare varianta, dar nu bag mana in foc pentru ea. Ideea e asa:
- prima data te intreaba de "sizeof(pv)" care este un sizeof(pointer) pentru ca pv este pointer iar pointerul are tot timpul dimensiune de 4 octeti;
- a doua oara te intreaba de sizeof(*pv) care (cred eu, destul de sigur) ca este dimesiunea primului elementt din vectorul ala, care este sizeof(char) = 1;
Obs: ca sa-ti ceara dimesiunea celui de-ale doilea elem din vector, de exemplu, ar fi trebuit sa scrie: sizeof((*pv)+1)

59.

Question 26 of 40 — 2.0 Points

The next Java example:

```
class Base{
public int vb;
public void Test(){System.out.print(" Test 1");}
}
class Subclass extends Base{
public void Test(){System.out.print(" Test 2");}
}
public class Main {
    public static void main(String[] args) {
        Base b = new Base();
        b.Test();
        Subclass d = new Subclass();
        d.Test();
        d = (Subclass) b;
        d.Test();

    }
}
```

asta e randul care-ti da eroare pentru ca incearca sa faca un obiect de tipul de baza sa devina un obiect de tipul derivat. gandeste-te ca obiectul derivat are si campuri + metode in plus fata de cel de baza , pe care nu are de unde sa le scoata;
- invers ar fi mers; daca vreau ca dintr-un obiect de derivat sa obtin unul de baza e ca si cand tai din el tot ce este in plus si raman doar cu campurile si metodele din clasa de baza.

prints:

- ( ) A. prints Test 1 Test 2 Test 1
- ( ) B. prints Test 1 Test 1 Test 1
- (●) C. the example generates a runtime ClassCastException because you can NOT do Downcasting
- ( ) D. prints Test 2 Test 2 Test 1
- ( ) E. prints Test 1 Test 2 Test 2

Reset Selection

Raspuns C - este la fel ca problema 21.

60.

Question 27 of 40 — 2.0 Points

When using a MessageDigest object to generate the hash value for a file you can do:

(1) use update method from MessageDigest to process a data block
(2) use digest method from MessageDigest to process a data block
(3) create a MessageDigest instance
(4) use digest method from MessageDigest to get the hash
(5) use update method from MessageDigest to get the hash

Which is the correct sequence:

- A. 3 + 5
- B. 3 + 2 + 5
- C. 3 + 1 + 2
- D. 3 + 1 + 4
- E. 3 + 4

Reset Selection

Raspuns : D

61.

Question 28 of 40 — 3.0 Points

For the macro-definition: `#define SQUARE(x) (x*(x))` if *x* has the value *5*, then the invocation `SQUARE (x+3)` generates the value:

- A. 29
- B. 19
- C. 43
- D. 23
- E. 64

E o mare smecherie cu aceasta intrebare, o idiotenie:

faci square (x+3*(x+3))=square (5 + 3* 8) = 29

Reset Selection

Raspuns A

62.

Question 29 of 40 — 4.0 Points

For the following program:

```
#include <stdio.h>
#include <malloc.h>
#include <openssl/md5.h>

int main(int argc, char **argv)
{
 if(argc == 2) {
     FILE* f = NULL;
     errno_t err;
     MD5_CTX ctx;
     unsigned char finalDigest[MD5_DIGEST_LENGTH];
     MD5_Init(&ctx);
     unsigned char* fileBuffer = NULL;
     err = fopen_s(&f, argv[1], "rb");
     if(err == 0) {
       fseek(f, 0, SEEK_END);
       int fileLen = ftell(f);
       fseek(f, 0, SEEK_SET);
       fileBuffer = (unsigned char*)malloc(fileLen);
       MD5_Update(&ctx, fileBuffer, fileLen);
       MD5_Final(finalDigest, &ctx);
      int count = 0;
      for( int i=0; i<MD5_DIGEST_LENGTH; i++) {
          printf( "%2X", finalDigest[i] );
      printf( " " );
      }
 fclose(f);
 }
}
else {
 printf("\n Usage Mode: ProgMainMD5.exe fSrc.txt");
 return 1;
    }
    return 0;
}
```

which of the following statements is complete and correct according to logical order of the source code:

A. The program opens file with the name received in the second command line parameter, establishes the length of the file to be processed, use the file md5.h from openssl folder, uses openSSL library, reads the file from HDD to RAM, executes MD5 hash function, allocates memory for RAM buffer, displays the content of hash, deallocates the memory of the buffer, closes the file

B. The program uses the MD5 library, opens a file with the name received in first command line parameter, establishes the length of the file to be processed, allocates memory for RAM buffer, reads the file from HDD to RAM, executes the MD5 hash function, displays the content of hash, closes file

C. The program uses the file md5.h from openssl folder, uses the openSSL library, opens a file with the name received in the second command line parameter, establishes the length of the output file, allocates memory for the buffer on HDD, reads the file from RAM to HDD, executes MD5 hash function, does not display the content of hash, deallocates memory of the buffer, reopens the file

D. The program uses the file md5.h from openssl folder, uses the openSSL library, opens a file with the name received in the second command line parameter, establishes the length of the file to be processed, allocates memory gor the RAM buffer, reads the file from HDD to RAM, executes the MD5 hash function, displays the content of hash, closes the file

E. The program uses the file md5.h from openssl folder, uses the openSSL library, opens a file with the name received the second command line parameter, establishes the length of the file to be processed, allocates memory for RAM buffer, reads the file from HDD to RAM, executes the MD5 hash function, displays the hash content, deallocates the memory of the buffer, closes the file

Raspuns :

63.

Question 30 of 40 — 1.0 Points

The polymorphism concept is implemented in Java:

A. using only functions overriding and the virtual mechanism

B. using only functions overloading and the virtual mechanism

C. using only functions overriding and the inheritance mechanism

D. using only functions overloading and the inheritance mechanism

E. using both functions overriding and overloading

Reset Selection

Asta e un subiect mai ciudat, dar ce trebuie sa retii tu este fix raspunsul la intrebarea asta: polimorfismul in Java se realizeaza prin suprascriere si supradefinire de metode.
- in java nu se pune accent pe mecanismul de metode virtuale, pentru ca , spre deosebire de C#, aici TOATE metodele sunt virtuale => se subintelege.

Aici raspunsurile sunt cam ciudate, asa ca nu bag mana in foc ca nu au gasit ei o chichita pe care sa o exploateze. Tu tb sa "keep things simple".

Raspuns : C, suprascriere(overriding) si mostenire

64.

Question 31 of 40 — 2.0 Points

The next example:

```
class Base{
public int vb;
public void Test(){System.out.print(" Test 1");}
}
class Subclass extends Base{
public void Test(){System.out.print(" Test 2");}
}
public class Main {
   public static void main(String[] args) {
      (1) Base b = new Base();
          b.Test();
      (2) Subclass d = new Subclass();
          d.Test();
      (3) b = d;
          b.Test();
   }
}
```

A. prints Test 1 Test 1 Test 1

B. prints Test 1 Test 2 Test 2

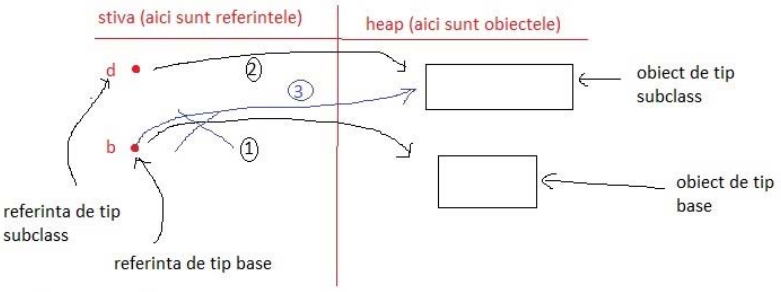
C. prints Test 1 Test 2 Test 1

D. the example generates a ClassCastException because you can NOT do Downcasting

E. prints Test 2 Test 2 Test 1

Reset Selection

Asta e fix cealalata situatie de care iti spuneam la intrebarea 26. Aici nu da eroare. In Java, nu conteaza tipul referintei, conteaza tipul obiectului, asa ca se va apela metodaa corecta in functie de tipul obiectului:



Raspuns B - similara cu problemele 21 si 59.

65.

Question 32 of 40 2.0 Points

The next class framework:

```
class Vehicle{
        protected String model;
        protected int cc;
        protected Vehicle(String M, int CC) { model = M; cc = CC; }
}
class Auto extends Vehicle{
        String series;
        public Auto(){
            super("Model",0);
            series = "0";
        }
        public Auto(String M, int CC, String S){
            model = M; cc = CC; series = S;
        }
}
```

Daca nu declaram niciun contructor in clasa de baza Vehicle, atunci compilatorul punea unul (metaforic vorbind :D) by default, fara parametri. Fiindca totusi noi avem un contructor aici cu parametri, compilatorul nu mai pune nimic, ne lasa pe noi sa fim "buni programatori" si sa ne descurcam singuri.

Astfel ca , in clasa derivata, constructorul cu parametri , desi nu este scris acolo, el incearca sa apeleze contructorul default din clasa de baza, pe care noi nu l-am pus. =>eroare de compilare

- A. generates compiler errors because the constructor with arguments from Auto is calling the default constructor from Vehicle
- B. generates compiler errors because the base class constructor is called with super("Model",0);
- C. generates compiler errors because the inheritance is implemented using implements and NOT extends
- D. the framework is correct defined, without compiler errors
- E. generates compiler errors because the Auto class is accessing inherited fields which are protected

Reset Selection

Raspuns A