LINK LA QUIZ

Question 1 - 35

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

O A. generates a public and private key for OSA over SHA-I hash

O B. it is initialized an "srandom" object using the Bouncy Castle provider

O C. generates a public and private key for RSA over MD5 hash

O D. none of these descriptions is correct

O E. it is initialized an "srandom" object using the default Java Sun provider

Question 2 -36

Which of the next Java statements it is NOT correct:

A. The deep-copy concept can be implemented by overriding clone method

B. The shallow-copy concept describes how the reference value is copied between two existing object

C. The deep-copy concept describes how the content is copied between two existing objects

D. The shallow-copy concept has NO effect on immutable objects

E. The deep-copy concept is implemented by default using operator between 2 objects

Question 3 - 38

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

O. abc abc true

abcd abcd true

E. abcd abc true

abcd abcd true

F. abcd abc false

abcd abc true

Question 4 - 40

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");
        Person p3 = p1;
        p3 = p2;
        Person p4 = p1;
        p1 = p2;
}
```

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 times, destructor – 4 times
C. Constructor – 2 times, copy constructor – 2 times, operator = 1 times, destructor – 4 times
D. Constructor – 3 times, copy constructor – 2 times, destructor – 5 times
E. Constructor – 2 times, copy constructor – 1 times, operator = 2 times, destructor – 2 times

Question 5

Using OpenSSL library to verify a RSA electronic signature, which of the following OpenSSL functions is not used 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

Question 6

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
```

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

Question 7

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 eligible for GC when the //other things line is reached ?

a. 2
b. 3
c. 5
d. 4
e. 1

Question 8

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
    }
}
```

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

O A. runtime exception

O B. 1

O C. compilation errors

0 D.3

O E.2

O F. it is not possible to determine

Question 9

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—ITGC Software Development, c—RO"
```

select the correct description

O A. Generates a public and private key store named "keystorelSM1.ks" that has the password "parolaks"

O B. Generates a public and private key store named "ISMCert1" that has the password "parolaks"

O C. Generates a RSA public and private key named "keystorelSM1.ks" that has the password "parolaks"

O D. Generates a RSA public and private key named "ISMCert1" that has the password "parolaks"

O E. Generates a public and private key store named "keystorelSM1.ks" that has the password "parolaisml"

Question 10

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”;

Question 11

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

Question 12

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 (Exception e) {
         System.out.println(e.getMessage());
         b = 33;
      } finally {
         a = 99;
      }
      System.out.println("a= " + a + " b= " + b + " c= " + c);
   }
}
```

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;

Question 13

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 ;
    }
}
```

O A. generates a compilation error because the ToString() method is accessing private fields

O B. generates a compilation error because the ToString() does NOT override a method inherited from Object

O C. generates a compilation error because the @Override annotation is used only for classes and NOT for methods

O D. generates a compilation error because the @Override annotations force us to extend explicitly the Object class

O E. The class is written correctly

Question 14

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());

    }
}
```

what is printed ?

O A. s1—Container 1 s2—Container 1

O B. s1—Container 2 s2—Container 2

O C. s1—Container 2 s2—Container 1

O D. compiler errors because the constructor is private

O E. s1—Nothing s2—Nothing

O F. s1—Container 1 s2—Container 2

Question 15

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);
    }
}
```

what is the result ?

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

Question 16

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?

O A. The private key is stored in parameter B

O B. The private key is stored in parameter D

O C. The private key is stored in parameter A

O D. Function RSA_private_decrypt is not define in OpenSSL library

O E. The private key has not to be specified in parameter list

Question 17

Please specify what the following program displays:

```
#include <iostream>
```

```
using namespace std;
class Car {
private:
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() << endl;
cout << c2.getProdYear() << " " << c2.getColor() << endl;
cout << c3.getProdYear() << " " << c3.getColor() << endl;
cout << c4.getProdYear() << " " << c4.getColor() << endl;
}
```

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;

Question 18

For the next statement:

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

what is the correct description

O A. It is created a DES cipher, used in ECB mode, with a given private key,using Bouncy Castle provider

O B. the instruction generates a NoSuchAlgorithmException at runtime

O C. It is created a AES cipher, used in ECB mode, with a given private key,using Bouncy Castle provider

O D. the instruction generates a NoSuchPaddingException at runtime

O E. It is created a DES cipher, used in CBC mode, with a given private key, using Sun provider

Question 19

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);
getchar();
}
```

O A. the values 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 are displayed;

O B. the values 02 4 6 8 are displayed;

O C. ambiguity error is reported at compiling time;

O D. the values 1 35 7 9 are displayed;

O E. ambiguity error is reported at run-time;

Question 20

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";
   }
}
```

O A. the constructor is ok and the class is defined without errors

O B. the constructor generates compiler errors because it changes the engine field value

O C. the constructor generates compiler errors because it changes the id field value

O D. the constructor generates compiler errors because it changes the noAutos static field and engine field values

O E. the constructor generates compiler errors because it changes the noAutos static field value

Question 21

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 + " b=  " + b + " c = " + c);
   }
```

O A. a=10 and b=2 , c=3

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

O C. a=10 and b=20 , c=3

O D. a=10 and b=20 , c=30

O E. a=30 and b=20 , c=30

Question 22

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. 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

Question 23

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:

O A. prints Test 1 Test 2 Test 1

O B. prints Test 1 Test 1 Test 1

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

O D. prints Test 2 Test 2 Test 1

O E. prints Test 1 Test 2 Test 2

Question 24

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

Question 25

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

Question 26

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, "src.txt", "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]);
}
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 HOD 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

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 HOD, reads the file from RAM to HOD, 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 for 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 HOD to RAM, executes the MD5 hash function, displays the hash content, deallocates the memory of the buffer, closes the file

Question 27

The polymorphism concept is implemented in Java:

O A. using only functions overriding and the virtual mechanism

O B. using only functions overloading and the virtual mechanism

O C. using only functions overriding and the inheritance mechanism

O D. using only functions overloading and the inheritance mechanism

O E. using both functions overriding and overloading

Question 28

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) {
      Base b = new Base();
      b.Test();
      Subclass d = new Subclass();
      d.Test();
      b = d;
      b.Test();
   }
}
```

O A. prints Test 1 Test 1 Test 1

O B. prints Test 1 Test 2 Test 2

O C. prints Test 1 Test 2 Test 1

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

O E. prints Test 2 Test 2 Test 1

Question 29

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;
   }
}
```

O A. generates compiler errors because the constructor with arguments from Auto is calling the default constructor from Vehicle

O B. generates compiler errors because the base class constructor is called with super("Model",0);

O C. generates compiler errors because the inheritance is implemented using implements and NOT extends

O D. the framework is correct defined, without compiler errors

O E. generates compiler errors because the Auto class is accessing inherited fields which are protected

Question 30
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. A compile-time error is generated;
B. A run-time error is generated;
C. The application displays 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200;
D. The application displays 200, 190, 180, 170, 160, 150, 140, 130, 120, 110, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100;
E. The application displays 200, 190, 180, 170, 160, 150, 140, 130, 120, 110, 100, 90, 80, 70, 60, 50, 40, 30, 20, 10;

Question 31
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 = ;

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

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

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

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

Question 32

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. A compile-time error is generated because the conversion derived object to base object is not allowed;
C. The program displays the strings: Polygon class, Rectangle class, Triangle class;
D. A run-time error is generated because the methods f are not declared as virtual in all defined classes;
E. The program displays the strings: Polygon class, Polygon class, Polygon class;

Question 33
In ASN1 V3 certificate structure, the OpenSSL function X509_set_version setting the X509 certificate version:
A. Writes value 2 in an INTEGER ASN1 Data Type;
B. Writes value 3 in a SEQUENCE ASN1 Data Type;
C. Writes value 3 in an INTEGER ASN1 Data Type;
D. Writes value 2 in the first byte of certificate;

E. Writes value 3 in an OBJECT IDENTIFIER ASN1 Data Type;
//See source code of ex37_GenerateX509.cpp

Question 34

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 has the same length with the private key;
B. sizeof(finalDigest) represents the digest of the message;
C. A compile-time error is generated because the function has other header;
D. RSA_PKCS1_PADDING is a parameter specifying the padding type;
E. RSA_PKCS1_PADDING is a parameter specifying the padding length;

Question 35

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. 50;
B. 0;

C. 5050;
D. The application generates a run-time error;
E. The source code cannot be compiled;

Question 36
In C++, a static attribute declared in a class is:
A. Always initialized in class definition;
B. A data definition when it is declared;
C. Used by class to manage the object collection;
D. Always defined as private attribute;
E. A member for each object having the class as data type;

Question 37
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 destructor method is defined in a wrong way;
B. A compile-time error is generated;
C. The attributes are defined in a wrong way;
D. The default constructor method is called 2 times;
E. A run-time error is generated;

```
//Compile error: Class “Car” has more than one default constructor
//The constructor with default values parameters is considered as default constructor but there is
//already a default constructor, so: compile error.
```

Question 38
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 destructor method;
B. A compile-time error is generated because of object assignment;
C. A compile-time error is generated because of object defining;
D. A run-time error is generated because of default constructor method;
E. The application runs properly;

Question 39
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);
        // cout << "constructor with parameters" << 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;
    }
    ~Person(){
        delete [] this->name;
        // cout << "destructor" << 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 << "assign"<<endl;
    }
};

Person Medie(Person a, Person b){  // copy constructor is called twice
    Person p;                   // default constructor is called
    p.age=(a.age+b.age)/2;
    return p;                    // copy constructor is called again
}

void main(){
    Person p1, p2(20, "John");    // default constructor is called for “p1”
                              // but because it is not defined,
                              // instead is called the constructor
                              // with default values parameters
    Person p3;                 // same as for “p1”
    p3 = Medie(p1, p2);          // assign (operator =) is called
}
```

Which of the following statements is CORRECT?

A. Operator = is called 2 times;

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

C. Copy constructor method is called 3 times;

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

E. Copy constructor is called 2 times;

Question 40

Given

```
class Base{
   static { System.out.print("b1 ");;}  // static init block – executed once
   { System.out.print("b2 ");;}        // instance init block – executed for every object created
   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. main b1 s1 b2 b3

B. main b2 b3 s1 b1 b2 b3

C. main b1 b3 s1 b2 b3

D. main b1 b2 b3 s1 b2 b3

E. main b1 b2 b3 b2 b3 s1

F. it can't be determined

G. b1 main b2 b3 s1 b2 b3

Question 41

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

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 compiler errors

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

E. the example generates runtime exceptions

// The constructor DataOutputStream(BufferedOutputStream) is undefined

Question 42
Being given the next class

```
public class Box {
   Object value;  // the value is typw Object so it will allow int and String values

   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");  // here is the hidden bug

int sum = 0;
for(Box box : numericalValues)
   sum += (int)box.getValue();  // for the 3rd object getValue will return a String
                              // that cannot be added to sum

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

select the correct affirmation:

A. The test will run without problems and it will print "The sum is 60"
B. The test will generate a run-time exception on line sum += (int)box.getValue();
C. The test will generate a compiler error on line numericalValues[0] = new Box(10);
D. The test will generate a compiler error on line sum += (int)box.getValue();
E. The test will generate a compiler error on line numericalValues[2] = new Box("30");

Question 43
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];  // it is created an array of 5 objects of Foo type
        System.out.println("Done !");
        // System.out.println(foos.length);    // <- THIS WILL PRINT HOW MANY OBJECTS// nu. asta arata
//cate referinte s-au creat in memorie, obiectele inca nu sunt create. singurul obiect creat( FOLOSIREA
//LUI NEW )  este acel array de Foo  }
}
```

How many objects are created before printing Done!

A. 5

B. 0

C. 1

D. 15

E. 6

F. 20

G. 16

H. 15

I. 21

Question 44

```
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. Compilation error
Implicit super constructor Parent() is undefined. Must explicitly invoke another constructor
B. PC
C. CP2
D. 2
E. PC2
F. CP

Question 45
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){ }// nu merge pt ca The method doSomething(int[]) in the type Base is not applicable for the arguments (int)
B. static void doSomething(int... args){ }(corecta)
C. static void doSomething(int... args, int a){ }// nu merge pt ca The variable argument type int of the method doSomething must be the last parameter
D. static void doSomething(int args...){ }// nu merge pt ca Syntax error on token "Invalid Character", delete this token
E. static void doSomething(int a, int... args){ }(corecta)

Question 46
Which one from the next Java statements it CORRECT:

A. abstract classes can be instantiated in objects; (slide 146)
B. abstract classes can contain non-abstract methods and instance variables;
C. All these statements are NOT correct in Java
D. interfaces can NOT be used as reference type; (slide 150)
E. interfaces can contain instance variables (slide 150)

Question 47
In the next sequence
How many objects are eligible for GC when line //other is reached?

```
class Something{
    Integer value = 200;
    Something doSomething(Something s){
        s = null;
        return s;  // this returns null
    }
    public static void main(){
        Something s1 = new Something();
        Something s2 = new Something();
        Something s3 = s1.doSomething(s2);  // it will get null from the return
        s1 = null;  // this is another one to get to garbage collector

        //other
    }
 }
```

A. Impossible to determine

B. 2

C. 1 ????????

D. 3

E. 0

Question 48

For the next code sequence

```
 class Student implements Person {
    public void Speak(){}
 }

 abstract class MasterStudent extends Student {}

 abstract class PhDStudent extends Student {
   public void Speak(String message){}
 }
class Graduate extends Student implements Person{
    public void Speak(){}
 }
```

what is the result ?

A. Compilation succeeds

B. Compilation error at line 9 - The hierarchy of the type PhDStudent is inconsistent

C. Compilation error at line 7 - The hierarchy of the type MasterStudent is inconsistent

D. Compilation error at line 3 - The type Person cannot be a superinterface of Student; a superinterface must be an interface

E. Compilation error at line 13 - Graduate: The hierarchy of the type Graduate is inconsistent

- Person: The type Person cannot be a superinterface of Graduate; a superinterface must be an interface

Question 49

The next sequence

```
public class Main {
   public static void main(String[] args) {
      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");
// this will create another object with different reference
      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)
// Integer values ranging from -128 to 127 are stored by the JVM in an area similar to String constant
pool because they are used in many situations.
         System.out.println("Integers are equal");
      else
         System.out.println("Integers are NOT equal");

      Integer i3 = 300;
      Integer i4 = 300;
      if (i3 == i4)
// because the integers are bigger than 127 (see above) then the references are different
         System.out.println("Integers are equal");
      else
         System.out.println("Integers are NOT equal");
```

```
    }
}
```

will print:

A. Strings are equal
Strings are NOT equal
Integers are equal
Integers are equal

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

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

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

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

F. Strings are NOT equal
Strings are equal
Integers are equal
Integers are NOT equal

Question 50
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);           // first call Base(5) constructor with params
        Subclass d = new Subclass(6);   // call Subclass(6) will:
                                // - call first Base() default constructor
                                // - call Subclass(6) constructor with params
    }
}
```

prints

A. BC call - SC call - DBC call

B. BC call - DBC call - SC call

C. BC call - BC call - SC call

D. BC call - SC call

E. DBC calln - BC call - SC call

Question 51

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;
    }
}
```

A. generates compiler errors because the Auto class is accessing inherited fields which are protected
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 constructor with arguments from Auto is calling the default constructor from Vehicle

```
//Compiler message:
//Implicit super constructor Main.Vehicle() is undefined. Must explicitly invoke another constructor
```

Question 52
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:

```
A.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