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
- ○ B. it is initialized an "srandom" object using the Bouncy Castle provider
- ○ C. generates a public and private key for RSA over MD5 hash
- ○ D. none of these descriptions is correct
- ○ E. it is initialized an "srandom" object using the default Java Sun provider

Reset Selection

Question 2 of 40 1.5 Points

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

Reset Selection

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();
        }
    }
}
```

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

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

Question 5 of 40 2.0 Points

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. main b1 s1 b2 b3
- B. b1 main b2 b3 s1 b2 b3
- C. main b1 b3 s1 b2 b3
- D. It can't be determined
- E. main s1 b1 b2 b3
- F. main b2 b3 s1 b1 b2 b3
- G. main b1 b2 b3 s1 b2 b3

Reset Selection

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");
 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 time, destructor – 4 times
- C. constructor – 2 times, copy constructor – 2 times, operator = 1 time, destructor – 4 times
- D. constructor – 3 times, copy constructor – 2 times, destructor – 5 times
- E. constructor – 2 times, copy constructor – 1 time, operator = 2 times, destructor – 2 times

Reset Selection

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

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
}
```

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

Question 9 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 eligible for GC when the `//other things` line is reached ?

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

Nu sunt sigur

Reset Selection

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
    }
}
```

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

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, c=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" Cred
- ( ) 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

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";

Reset Selection

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

Reset Selection

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);
    }
}
```

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

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
- B. generates a compilation error because the ToString() does NOT override a method inherited from Object
- C. generates a compilation error because the @Override annotation is used only for classes and NOT for methods
- D. generates a compilation error because the @Override annotations force us to extend explicitly the Object class
- E. The class is written correctly

Reset Selection

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);
        Subclass d = new Subclass(6);
    }
}
```

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

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());
    }
}
```

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

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);
    }
}
```

what is the result ?

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

Reset Selection

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

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 ( )<<" ; ";
}
```

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

Nu sunt sigur

Reset Selection

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);
}
```

Daca nu e gresit aici. Presupun ca al doilea parametru e v, nu vect

- 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

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";
    }
}
```

- 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

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);

}
```

generates the results: CRED ??

- 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

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

Nu am nici o idee de ce. Puteti explica? Am rulat si asta e rezultatul.

Reset Selection

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();

    }
}
```

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

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

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

Reset Selection

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

Reset Selection

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

Nu sunt sigur

Reset Selection

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

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;
        }
}
```

- 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