PEMROGRAMAN BERORIENTASI OBJEK SI-46-07 [FNO]

Dashboard / My courses / FAKULTAS REKAYASA INDUSTRI (FRI) / PRODI S1 SISTEM INFORMASI (FRI) / 2324/1 / ISI2A3-SI-46-07 / UTS / SOAL UTS

Started on	Tuesday, 7 November 2023, 10:05 AM
State	Finished
Completed on	Tuesday, 7 November 2023, 11:34 AM
Time taken	1 hour 28 mins
Marks	43.17/50.00
Grade	**8.63** out of 10.00 (**86**%)

Question 1

Correct

Mark 5.00 out of 5.00

From a project to Create a simple Student Management System that allows you to add and manage student records. Each student record should have the following information:

- Student ID (a unique identifier)
- Name
- Age
- Course (e.g., "Computer Science," "Mathematics," etc.)

Create a `Student` class with the necessary attributes (Student ID, Name, Age, Course), a constructor to initialize these attributes, and getter and setter methods for each attribute.

```
class ✓ Student {
  private ✓ int studentID;
  private String name;
  private int age;
  private String course;

  public Student ✓ (int studentID, String name, int age, String course) {
    this.studentID = studentID;
    this.name = name;
    this.age = age;
    this.course = course;
  }

  public int getStudentID() {
    return studentID ✓ ;
  }

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this ✓ .name = name;
  }

  public int getAge() {
    return age;
  }

  public void setAge(int age) {
    this.age = age;
  }

  public String getCourse() {
    return course;
  }
```

```
    public void setCourse(String course) {
        this.course = course;
    }
}
```

Your answer is correct.

The correct answer is:

From a project to Create a simple Student Management System that allows you to add and manage student records. Each student record should have the following information:

- Student ID (a unique identifier)
- Name
- Age
- Course (e.g., "Computer Science," "Mathematics," etc.)

Create a `Student` class with the necessary attributes (Student ID, Name, Age, Course), a constructor to initialize these attributes, and getter and setter methods for each attribute.

```
[class] Student {
    [private]  int studentID;
    private String name;
    private int age;
    private String course;

    public [Student] (int studentID, String name, int age, String course) {
        this.studentID = studentID;
        this.name = name;
        this.age = age;
        this.course = course;
    }

    public int getStudentID() {
        return [studentID];
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        [this].name = name;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getCourse() {
        return course;
    }

    public void setCourse(String course) {
        this.course = course;
    }
}
```

Question **2**

Correct

Mark 5.00 out of 5.00

From a Project: Create a simple Student Management System that allows you to add and manage student records. Each student record should have the following information:

- Student ID (a unique identifier)
- Name
- Age
- Course (e.g., "Computer Science," "Mathematics," etc.)

Create a `StudentDatabase` class that uses an array to store `Student` objects. Implement the following methods:

- `addStudent(Student student)`: Add a student to the database.
- `removeStudent(int studentID)`: Remove a student from the database by their ID.

Here is the snippet of the program

```
class StudentDatabase {
  private Student[] students;
  private int currentIndex;
  public StudentDatabase(int capacity) { // membuat database student sebesar maksimumnya --> capacity
    students = new Student[capacity];  // constructor untuk membuat object student dari class Student
    currentIndex = [ 0 ] ✓ ; // perhatikan bahwa index dimulai dari nol
  }
  public void addStudent([ Student ] ✓ student) {
    if (currentIndex < students.[ length ] ✓ ) { // jika jml student masih belum maksimum
      students[currentIndex] = student;
      [currentIndex] ✓ ++;
    } else {
      System.out.println("Database is full. Cannot add more students.");
    }
  }

  public void removeStudent(int studentID) {
    for (int i = 0; i < currentIndex; i++) {
      if (students[i].getStudentID() == studentID) {
        students[i] = null;
        // Shift remaining elements to fill the gap
        for (int j = i; j < currentIndex - 1; j++) {
          students[j] = students[j + 1];
        }
        currentIndex--;
        System.out.println("Student with ID " + studentID + " removed from the database.");
        return;
      }
    }
    System.out.println("Student with ID " + studentID + " not found in the database.");
  }
}
```

Your answer is correct.

The correct answer is:

From a Project: Create a simple Student Management System that allows you to add and manage student records. Each student record should have the following information:

- Student ID (a unique identifier)
- Name
- Age
- Course (e.g., "Computer Science," "Mathematics," etc.)

Create a `StudentDatabase` class that uses an array to store `Student` objects. Implement the following methods:

- `addStudent(Student student)`: Add a student to the database.
- `removeStudent(int studentID)`: Remove a student from the database by their ID.

Here is the snippet of the program

```
class StudentDatabase {
    private Student[] students;
    private int currentIndex;
    public StudentDatabase(int capacity) { // membuat database student sebesar maksimumnya --> capacity
        students = new Student[capacity];  // constructor untuk membuat object student dari class Student
        currentIndex = [0]; // perhatikan bahwa index dimulai dari nol
    }
    public void addStudent([Student] student) {
        if (currentIndex < students.[length]) { // jika jml student masih belum maksimum
            students[currentIndex] = student;
            [currentIndex]++;
        } else {
            System.out.println("Database is full. Cannot add more students.");
        }
    }

    public void removeStudent(int studentID) {
        for (int i = 0; i < currentIndex; i++) {
            if (students[i].getStudentID() == studentID) {
                students[i] = null;
                // Shift remaining elements to fill the gap
                for (int j = i; j < currentIndex - 1; j++) {
                    students[j] = students[j + 1];
                }
                currentIndex--;
                System.out.println("Student with ID " + studentID + " removed from the database.");
                return;
            }
        }
        System.out.println("Student with ID " + studentID + " not found in the database.");
    }
}
```

Question **3**

Correct

Mark 5.00 out of 5.00

from a project:

Create a simple Student Management System that allows you to add and manage student records. Each student record should have the following information:

- Student ID (a unique identifier)
- Name
- Age
- Course (e.g., "Computer Science," "Mathematics," etc.)

Implement the following methods:

- `displayAllStudents()`: Display all students in the database.

assuming that all required classes and methods needed is already made for you

```
public void displayAllStudents() {
      System.out.println("List of Students:");
      for (int i = 0; I < [currentIndex] ✓ ;[i++] ✓ ) {
         [Students] ✓ student = students[i];
         System.out.println("ID: " + student.[getStudentID()] ✓ + ", Name: " + student.getName() + ", Age: " + student.
[getAge()] ✓ + ", Course: " + student.getCourse());
      }
   }
```

Your answer is correct.

The correct answer is:
from a project:

Create a simple Student Management System that allows you to add and manage student records. Each student record should have the following information:

- Student ID (a unique identifier)
- Name
- Age
- Course (e.g., "Computer Science," "Mathematics," etc.)

Implement the following methods:

- `displayAllStudents()`: Display all students in the database.

assuming that all required classes and methods needed is already made for you

```
public void displayAllStudents() {
      System.out.println("List of Students:");
      for (int i = 0; I < [currentIndex];[i++]) {
         [Students] student = students[i];
         System.out.println("ID: " + student.[getStudentID()] + ", Name: " + student.getName() + ", Age: " + student.
[getAge()] + ", Course: " + student.getCourse());
      }
   }
```

Question **4**

Correct

Mark 5.00 out of 5.00

From a project: Create a `StudentDatabase` class that uses an array to store `Student` objects. Implement the following methods:

- `findStudent(int studentID)`: Find and return a student by their ID.
- assuming that class Student and StudentDatabase is already made for you

```
public  Student ✓ findStudent(int  studentID ✓ ) {
  for (int i = 0; i < currentIndex; i++) {
    if (students[I]. getStudentID() ✓ == studentID) {
      return  students[i] ✓ ;
    }
  }
  return null;
}
```

Your answer is correct.

The correct answer is:
From a project: Create a `StudentDatabase` class that uses an array to store `Student` objects. Implement the following methods:

- `findStudent(int studentID)`: Find and return a student by their ID.
- assuming that class Student and StudentDatabase is already made for you

```
public [Student]findStudent(int [studentID]) {
  for (int i = 0; i < currentIndex; i++) {
    if (students[I].[getStudentID()] == studentID) {
      return [students[i]];
    }
  }
  return null;
}
```

Question **5**

Correct

Mark 5.00 out of 5.00

```
// Kelas Mahasiswa
class Mahasiswa ✓ {
   private String nim;
   private String nama;
   private int usia;

   public Mahasiswa(String nim, String nama, int usia) {
      this.nim = nim;
      this ✓ .nama = nama;
      this.usia = usia;
   }

   public void displayInfo() {
      System.out.println("NIM: " + nim);
      System.out.println("Nama: " + nama);
      System.out.println("Usia: " + usia);
   }
}

// Kelas Dosen
class Dosen {
   private String nip;
   private String nama;
   private String mataKuliah;

   public Dosen(String nip, String nama, String mataKuliah) {
      this.nip = nip;
      this.nama = nama;
      this.mataKuliah = mataKuliah;
   }

   public void displayInfo() {
      System.out.println("NIP: " + nip);
      System.out.println("Nama Dosen: " + nama);
      System.out.println("Mata Kuliah: " + mataKuliah);
   }
}

// Kelas utama untuk menguji sistem informasi mahasiswa
public class SistemInformasiMahasiswa {
   public static void main(String[] args) {
      // Membuat objek mahasiswa
      Mahasiswa mahasiswa1 = new ✓ Mahasiswa("123456", "Jonih ", 20);
```

```
        // Menampilkan informasi mahasiswa
        System.out.println("Informasi Mahasiswa:");
        mahasiswa1.displayInfo();
        System.out.println();

        // Membuat objek dosen
        [ Dosen ] ✓ dosen1 = new Dosen("7890", "DR. Baek Hati", "Pemrograman Java");

        // Menampilkan informasi dosen
        System.out.println("Informasi Dosen:");
        dosen1.[ displayInfo ] ✓ ();
    }
}
```

Your answer is correct.

The correct answer is:

```
// Kelas Mahasiswa
class [Mahasiswa]{
    private String nim;
    private String nama;
    private int usia;

    public Mahasiswa(String nim, String nama, int usia) {
        this.nim = nim;
        [this].nama = nama;
        this.usia = usia;
    }

    public void displayInfo() {
        System.out.println("NIM: " + nim);
        System.out.println("Nama: " + nama);
        System.out.println("Usia: " + usia);
    }
}

// Kelas Dosen
class Dosen {
    private String nip;
    private String nama;
    private String mataKuliah;

    public Dosen(String nip, String nama, String mataKuliah) {
        this.nip = nip;
        this.nama = nama;
        this.mataKuliah = mataKuliah;
    }

    public void displayInfo() {
        System.out.println("NIP: " + nip);
        System.out.println("Nama Dosen: " + nama);
        System.out.println("Mata Kuliah: " + mataKuliah);
    }
}

// Kelas utama untuk menguji sistem informasi mahasiswa
public class SistemInformasiMahasiswa {
    public static void main(String[] args) {
        // Membuat objek mahasiswa
        Mahasiswa mahasiswa1 = [new] Mahasiswa("123456", "Jonih ", 20);

        // Menampilkan informasi mahasiswa
        System.out.println("Informasi Mahasiswa:");
        mahasiswa1.displayInfo();
        System.out.println();

        // Membuat objek dosen
        [Dosen] dosen1 = new Dosen("7890", "DR. Baek Hati", "Pemrograman Java");

        // Menampilkan informasi dosen
        System.out.println("Informasi Dosen:");
        dosen1.[displayInfo]();
    }
}
```

Question **6**

Correct

Mark 1.00 out of 1.00

Dalam perkuliahan kita bisa pandang [Mahasiswa] ✓ sebagai Class dan [Nur Pamudji] ✓ adalah salah satu obyek, demikian pula dalam hal hewan [Burung beo di rumah Oom Sule] ✓ adalah obyek dan Class nya adalah [Burung] ✓

Your answer is correct.

The correct answer is:
Dalam perkuliahan kita bisa pandang [Mahasiswa] sebagai Class dan [Nur Pamudji] adalah salah satu obyek, demikian pula dalam hal hewan [Burung beo di rumah Oom Sule] adalah obyek dan Class nya adalah [Burung]

Question **7**

Correct

Mark 1.00 out of 1.00

```
import math

class Circle:
    def __init__(self, radius):
        self.radius = radius

    def hitung_luas(self):
        luas = math.pi * self.radius**2
        return luas

    def hitung_keliling(self):
        keliling = 2 * math.pi * self.radius
        return keliling

jari_jari = float(input("Masukkan jari-jari lingkaran: "))
lingkaran = Circle(jari_jari)

luas = lingkaran.hitung_luas()
keliling = lingkaran.hitung_keliling()

print(f"Luas lingkaran: {luas}")
print(f"Keliling lingkaran: {keliling}")
// Dari program di atas maka  hitung_luas dan hitung_keliling adalah method yang disediakan oleh class Circle
```

Select one:

● True ✓

○ False

The correct answer is 'True'.

Question **8**

Correct

Mark 1.00 out of 1.00

An object is an instantiation of a class. Pernyataan ini dapat diterjemahkan sebagai -sebuah obyek adalah sebuah perwujudan dari sebuah kelas.

Select one:

● True ✓

○ False

The correct answer is 'True'.

Question **9**

Correct

Mark 1.00 out of 1.00

A Class can be viewed as a template or definition or blueprint of a group of objects.

For example: a banana class. Its objects could be:

- the banana being eaten by Hawari
- the banana desired by Alicia.

Select one:

● True ✓

○ False

The correct answer is 'True'.

Question **10**

Partially correct

Mark 0.50 out of 1.00

Which of the following variable definitions are legitimate if it is an instance variable (not class variable)? (choose all that apply)

Select one or more:

□ A.

protected int a;

□ B.

private static final int a;

☑ C.

public int a = 1; ✓

□ D.

public static int a;

Your answer is partially correct.

You have correctly selected 1.
The correct answers are:

protected int a;,

public int a = 1;

Question **11**

Partially correct

Mark 0.67 out of 1.00

Which one is valid main method that we can use to start a class from the command line? (choose all that apply)

Select one or more:

☑ A.

static public void main(String[] args) { } ✓

□ B.

public static int main(String[] args) { }

□ C.

public static void main(String[] test) { }

☑ D.

public static void main(String[] args) { } ✓

Your answer is partially correct.

You have correctly selected 2.
The correct answers are:

public static void main(String[] args) { },

static public void main(String[] args) { },

public static void main(String[] test) { }

Question **12**

Correct

Mark 1.00 out of 1.00

What is the output of this code going to be?

```
public class Test {
        public int t = 4;
        public static void main (String [] args) {
                new Test().NumberPlay();
        }

        public void NumberPlay() {
                int t = 2;
                t = t + 5;
                this.t = this.t - 2;
                t = t - this.t;
                System.out.print(t + " ");
                System.out.println(this.t);
         }
}
```

Select one or more:

□ A.

0 -9

□ B.

9 0

□ C.
2 7

□ D.

7 2

□ E.

2 5

☑ F.

5 2 ✓

Your answer is correct.

The correct answer is:

5 2

Question **13**

Incorrect

Mark 0.00 out of 1.00

The following source file is given the name FirstClass.java

```
import java.*;
public class FirstClass { }
class SecondClass { }
public class ThirdClass { }
```

What error will the compiler will likely generate?

Select one or more:

□ A.
class SecondClass must be public

□ B.
public class FirstClass could not be public

□ C.
public class SecondClass must be defined in other file named "SecondClass.java" because there cannot be more than one class with a public modifier in the same file

☑ D.
package java not found in import ✖

Your answer is incorrect.

The correct answer is:
public class SecondClass must be defined in other file named "SecondClass.java" because there cannot be more than one class with a public modifier in the same file

Question **14**

Incorrect

Mark 0.00 out of 1.00

We are given an array named arr as follow:

```
public class TestClass {
    int[][] arr = { { 1, 2, 3 }, { 4, 5 } };

    static public void main(String[] args) {
            TestClass test = new TestClass();
            // print the value of 5 from that array
    }
}
```

What code should be put to print the value of 5 from the array named arr?

Select one or more:

□ A.
System.out.println(test.arr[1][1]);

☑ B.
System.out.println(test.arr[0][1]); ✖

□ C.
System.out.println(test.arr[1][0]);

Your answer is incorrect.

The correct answer is:
System.out.println(test.arr[1][1]);

Question **15**

Correct

Mark 1.00 out of 1.00

We are given a two-dimensional array named arr as follow:

```
public class TestClass {
      int[][] arr = { { 1, 2, 3 }, { 4, 5 } };

      static public void main(String[] args) {
                TestClass test = new TestClass();
                // print the length of the second element of the first dimension of the array arr
      }
}
```

What code should be put to print the length of the second element of the first dimension of the array named arr, which is { 4, 5 }?

Select one or more:

☑ A.
System.out.println(test.arr[1].length); ✓

□ B.
System.out.println(test.arr[0].length);

□ C.
System.out.println(test.arr.length);

□ D.
System.out.println(test.arr[1].length());

Your answer is correct.

The correct answer is:
System.out.println(test.arr[1].length);

Question **16**

Incorrect

Mark 0.00 out of 1.00

If you try to compile and run this class, what will happen?

```
public class LocalTest {
    public static void main(String[] args) {
            int i;
            i = 20;
            System.out.print("int i = " + (++i));
    }
}
```

Select one or more:

□ A.
It will compile and output 21 to the screen

☑ B.
It will compile, but will produce interpreter error when executed ✖

□ C.
It will compile and output 20 to the screen

☑ D.
It will not compile and give the compile error ✖

Your answer is incorrect.

The correct answer is:
It will compile and output 21 to the screen

Question **17**

Correct

Mark 1.00 out of 1.00

Suppose the Main class is located in the package main, and the Student class is located in the package model

Analyze this code:

```
// located in the folder main
package main;

import model.Student;
public class Main {
    public static void main(String[] args) {
            Student student = new Student();
            student.setName("Toni");
            System.out.println(student.getName());
    }
}

// located in the different folder named model
package model;

class Student {
    private String name;
    private String nim;

    public void setName(String name) {
            this.name = name;
    }

    public void setNim(String nim) {
            this.nim = nim;
    }

    public String getName() {
            return this.name;
    }

    public String getNim() {
            return this.nim;
    }
}
```

What will be the output?

Select one or more:

☑ A.

The code will compile and run fine ✗

☑ B.
The code will not compile because the Student class located in package model could not be accessed from any class outside its package ✓

☐ C.
The code will not compile because the Main class should not have public modifier

☐ D.
The code will compile but have runtime error

Your answer is correct.

The correct answer is:
The code will not compile because the Student class located in package model could not be accessed from any class outside its package

Question **18**

Correct

Mark 1.00 out of 1.00

When a class B, have access modifier "protected", it means

Select one or more:

☐ A.
It cannot be accessed any where

☐ B.
It can be accessed any where

☐ C.
It can be accessed only in its package

☑ D.
It can be accessed in its package and its child class that extend class B ✓

Your answer is correct.

The correct answer is:
It can be accessed in its package and its child class that extend class B

Question **19**

Correct

Mark 1.00 out of 1.00

Someone has just finished creating a class and wants you to review it:

```
class DataServer extends Server{
      public String serverName;
      public DataServer () {
                  serverName = "Customer Service";
                  super(serverName);
      }
}
```

What do you tell the programmer:

Select one or more:

□ A.
The code will compile and run fine

☑ B.
The code will not compile because there is something wrong in the DataServer method ✓

□ C.
The code will compile but the error will occurred when the class is instantiated

Your answer is correct.

The correct answer is:
The code will not compile because there is something wrong in the DataServer method

Question **20**

Correct

Mark 1.00 out of 1.00

Examine the following code:

```
public class CheckDefault {
      private static boolean test;
      public static void showTest() {
                System.out.println(test);
      }
}
```

If we invoke the static method showTest(), what will the output be?

Select one or more:

☑ A.
true ✗

☑ B.
false ✓

☑ C.
none, the class will not compile ✗

Your answer is correct.

The correct answer is:
false

Question **21**

Incorrect

Mark 0.00 out of 1.00

Suppose we have two classes as follow:

```
// in the folder payroll
package payroll;
public class UserConnection{
      protected int users = 0;
      public static String getServerName() {
                  return "Main Server";
      }
}

// in the folder client
package client;
import payroll.UserConnection;
public class ServerConnection {
      UserConnection u = new UserConnection();
      public int getUsers() {
                  return u.users;
      }
}

// in the folder main
package main;
import client.ServerConnection;
public class Main {
public static void main(String[] args) {
            ServerConnection s = new ServerConnection();
}
}
```

What will happen when these classes are compiled and run?

Select one or more:

☑ A.
The file with class UserConnection will not compile ✖

□ B.
The files will compile and run without problem

□ C.
The file with class Main will not compile

☑ D.
The files will compile, but an error will occur in the runtime ✖

□ E.
The file with class ServerConnection will not compile

Your answer is incorrect.

The correct answer is:
The file with class ServerConnection will not compile

Question **22**

Correct

Mark 1.00 out of 1.00

Examine the following code:

```
class TestServer {
      public TestServer() {
                 int users = 1;
      }

      public void increment() {
                 users = users + 1;
      }

      public static void main(String [] args){
                 increment();
                 System.out.println("Variable users = " + users);
      }
}
```

What will be the output of this class will be when compile and run?

Select one or more:

☐ A.
Output is 1

☐ B.
Output is 2

☐ C.
The file will compile, but will give an error when run

☑ D.
The file will not compile because we can’t directly access method increment inside the static method (the object should be instantiated first then use the object to access increment method) ✓

Your answer is correct.

The correct answer is:
The file will not compile because we can’t directly access method increment inside the static method (the object should be instantiated first then use the object to access increment method)

Question **23**

Correct

Mark 1.00 out of 1.00

Examine the following code:

```
class TestServer {
      public TestServer() {
            int users = 1;
      }

      public void increment() {
            users = users + 1;
      }

      public static void main(String [] args){
            increment();
            System.out.println("Variable users = " + users);
      }
}
```

What will be the output of this class will be when compile and run?

Select one or more:

☑ A.

Output is 2 ✗

☑ B.

The file will not compile because we can’t directly access method increment inside the static method (the object should be instantiated first then use the object to access increment method) ✓

☐ C.

Output is 1

☐ D.

The file will compile, but will give an error when run

Your answer is correct.

The correct answer is:

The file will not compile because we can’t directly access method increment inside the static method (the object should be instantiated first then use the object to access increment method)

Question **24**

Correct

Mark 1.00 out of 1.00

Examine the following code:

```
class TestServer {
      public TestServer() {
            int users = 1;
      }

      public void increment() {
            users = users + 1;
      }

      public static void main(String [] args){
            increment();
            System.out.println("Variable users = " + users);
      }
}
```

What will be the output of this class will be when compile and run?

Select one or more:

☑ A.
Output is 2 ✗

□ B.
Output is 1

☑ C.
The file will not compile because we can't directly access method increment inside the static method (the object should be instantiated first then use the object to access increment method) ✓

□ D.
The file will compile, but will give an error when run

Your answer is correct.

The correct answer is:
The file will not compile because we can't directly access method increment inside the static method (the object should be instantiated first then use the object to access increment method)

Question **25**

Correct

Mark 1.00 out of 1.00

Examine this code:

```
public class  A {
       public static void main(String[] args) {
              Calculation c = new Calculation();
              c.print1();
       }
}

class Calculation {
       private int number;

       public void setNumber(int number) {
              this.number = number;
       }

       public void print1() {
              int sum = 0;
              for (int i=0; i<this.number; i++) {
                     if ( i % 2 == 0)
                            sum += i;
              }
              System.out.println("Sum is " + sum);
       }
}
```

What will be the output when the code is compiled and run?

Select one or more:

☐ A.
null

☑ B. 0 ✓

☐ C.
20

☑ D.
The code will not compile ✗

☐ E.
10

Your answer is correct.

The correct answer is: 0

Question **26**

Correct

Mark 1.00 out of 1.00

Examine this code:

```
public class  A {
      public static void main(String[] args) {
                  Calculation c = new Calculation();
                  c.setNumber(10);
                  c.print1();
      }
}

class Calculation {
      private int number;

      public void setNumber(int number) {
                  this.number = number;
      }

      public void print1() {
                  int sum = 0;
                  for (int i=0; i<this.number; i++) {
                              if ( i % 2 == 0)
                                          sum += i;
                  }
                  System.out.println("Sum is " + sum);
      }
}
```

What will be the output when the code is compiled and run?

Select one or more:

□ A.
null

□ B.
0

□ C.
The code will not compile

□ D.
10

☑ E.
20 ✓

Your answer is correct.

The correct answer is:

20

Question **27**

Correct

Mark 1.00 out of 1.00

Examine this code:

```
public class  A {
      public static void main(String[] args) {
              Calculation c = new Calculation();
              c.setNumber(10);
              c.print1();
      }
}

class Calculation {
      private int number;

      public void setNumber(int number) {
              this.number = number;
      }

      public void print1() {
              int sum = 0;
              for (int i=0; i<this.number; i++) {
                      if ( i % 2 == 0)
                              break;
                      sum += i;
              }
              System.out.println("Sum is " + sum);
      }
}
```

What will be the output when the code is compiled and run?

Select one or more:

□ A.
10

□ B.
20

□ C.
The code will not compile

☑ D.
0 ✓

□ E.
null

Your answer is correct.

The correct answer is:
0

Question **28**

Incorrect

Mark 0.00 out of 1.00

Examine this code:

```
public class  A {
      public static void main(String[] args) {
               Calculation c = new Calculation();
               c.setNumber(10);
               c.print1();
      }
}

class Calculation {
      private int number;

      public void setNumber(int number) {
               this.number = number;
      }

      public void print1() {
               int sum = 0;
               for (int i=0; i<this.number; i++) {
                        if ( i % 2 == 0)
                                 continue;
                        sum += i;
               }
               System.out.println("Sum is " + sum);
      }
}
```

What will be the output when the code is compiled and run?

Select one or more:

□ A. 15

□ B.
25

□ C.
null

☑ D.
The code will not compile ✖

□ E.
5

Your answer is incorrect.

The correct answer is:

25

Question **29**

Incorrect

Mark 0.00 out of 1.00

This class was programmed to keep track of inventory using a special numbering system. The convertCode() method creates a unique inventory number:

```
class Inventory{
    public static void main (String [] args){
        int code = 237;
        convertCode(code); // *
    }

    public static void convertCode(int inv){
        inv = inv + 100000;
    }
}
```

What will the variable code equal after the method convertCode() is run?

Select one or more:

□ A.
237

☑ B.
100237 ✖

☑ C.
The code will not compile ✖

Your answer is incorrect.

The correct answer is:
237

Question **30**

Correct

Mark 1.00 out of 1.00

Examine the following code:

```
class ReferenceTest {
    public static void main (String [] args) {
        int a = 1;
        ReferenceTest rt = new ReferenceTest();
        System.out.print(a);
        rt.modify(a);
        System.out.print(a);
    }

    void modify(int number) {
        number = number + 1;
        System.out.print(number);
    }
}
```

What will be the output when it compile and run?

Select one:

● A.
121 ✓

○ B.
122

○ C.
Nothing, this class will not compile

○ D.
111

○ E.
100

Your answer is correct.

The correct answer is:
121

Previous Activity

Jump to...