

**Nama : Muhammad Fadil Amin**

**NIM : 20210040113**

**Kelas : TI21E**

**Pemrograman Berorientasi Objek-Nilai Tambahan**

1. Urutan keyword yang digunakan untuk mendeklarasikan sebuah class menggunakan JAVA dari atas ke bawah adalah ....
  - a. Class -> Import -> Package
  - b. Package -> Import -> Class**
  - c. Import -> Package -> Class
  - d. Package -> Class -> Import
  - e. Import -> Class -> Package
2. Java mendukung fitur yang memudahkan programmer untuk membuat kode program yaitu...
  - a. Pointer
  - b. Structure
  - c. Header File
  - d. Multiple Inheritance**
  - e. Interface
3. Pengertian dari istilah Objek pada bahasa pemrograman JAVA adalah ...
  - a. Anggota dari class
  - b. Instance dari class**
  - c. Blue print dari class
  - d. Class yang mempunyai fungsi main
  - e. Class yang mempunyai konstruktor
4. Manakah yang benar dalam pembuatan class berikut ini:

```
-----  
class Computer{  
    String brand;  
    int size_in_inch;  
    void infoComputer(){  
        System.out.println("Brand :" + this.brand);  
        System.out.println("Monitor size :" + this.size_in_inch);  
    }  
}
```

```
-----  
class Laptop {  
    String brand;  
    Laptop(String brand){  
        this.brand = brand;  
    }  
  
    public String infoLaptop(){  
        System.out.println("Brand:" + this.brand);  
    }  
  
    public void setBrand(String brang){  
        this.brand = brand;  
    }  
}
```

```
}
```

```
-----
private class Motor{
    String brand;
    Motor(String brand){
        this.brand = brand;
    }

    public String infoMotor(){
        return "Brand:" + this.brand;
    }

    public void setBrand(String brand){
        this.brand = brand;
    }
}
-----
```

```
class Room{
    int capacity;
    String name;

    Room(){

    }

    void setRoomData(int capacity, String name, String location){
        this.capacity = capacity;
        this.name = name;
    }
}
```

- a. Computer dan Room
  - b. Computer, Laptop dan Motor
  - c. Laptop dan Motor
  - d. Laptop, Motor dan Room
  - e. **Computer**
5. Header method manakah yang termasuk dalam jenis setter (mutator)?
- a. **public String setData(String name)**
  - b. double setNumber(int number)
  - c. protected Boolean setTrueOrFalse(boolean data)
  - d. int getValue(int value)
  - e. void getData(int data)
6. Method yang digunakan untuk memberikan nilai pada variabel atau argumen dikenal dengan istilah
- a. **Setter**
  - b. Getter
  - c. Giver
  - d. Deliver
  - e. Letter
7. Method yang digunakan untuk memberikan nilai pada variabel atau argumen dikenal dengan istilah
- a. **Setter**
  - b. Getter
  - c. Giver

- d. Deliver
  - e. Letter
8. Manakah penulisan yang benar dari kode program berikut ini, jika sebuah class Komputer adalah turunan dari class PerangkatElektronik
- a. class PerangkatElektronik extends Komputer
  - b. class PerangkatElektronik implements Komputer
  - c. class Komputer extends PerangkatElektronik**
  - d. class Komputer implements PerangkatElektronik
  - e. Tidak ada jawaban yang benar
9. Konstruktor adalah
- a. Nama method sama dengan nama class**
  - b. Method yang hanya dipanggil 1x
  - c. Method yang dipanggil ketika proses penciptaan objek
  - d. Method yang digunakan untuk melakukan proses inisialisasi nilai
  - e. Semua Benar
10. Manakah bentuk penulisan kode program berikut ini yang tepat?
- a. class Book extends Paper, Novel, Newspaper implements Magazine
  - b. class Book Implements Paper extends Novel, Newspaper, Magazine
  - c. class Book implements Paper, Novel, Newspaper extends Magazine
  - d. [class Book extends Paper implements Novel, Newspaper, Magazine ]**
  - e. class Book extends Paper implements Novel, implements Newspaper, implements Magazine
11. Konsep PBO pada JAVA tidak mendukung multiple inheritance atau pewarisan banyak orang tua. Untuk menangani masalah tersebut, maka JAVA menyediakan fitur atau konsep lain yaitu
- a. Constructor
  - b. Abstract
  - c. Polymorphism**
  - d. Interface
  - e. OverCoding
12. Jika terdapat method seperti berikut ini:

```
public double getResultMultiply(int a, int b){
    return a * b;
}
```

maka bentuk dari method overloading yang benar adalah ..

- a. public double getResultMultiply(double a, double b)
  - b. public double getResultMultiply(int a, int b, int c)
  - c. public double getResultMultiply(int a, double b)
  - d. public double getResultMultiply(double a, int b)
  - e. Semua Benar**
13. Sebuah variabel atau atribut dibuat menggunakan modifier private dengan alasan keamanan program, sehingga agar variabel tersebut dapat diakses dikelas lain dibuatlah method setter dan getter. Pernyataan tersebut adalah salah satu konsep PBO yaitu :
- a. Polymorfisme
  - b. Encapsulation**
  - c. Abstract
  - d. Interface
  - e. Package

14. Output apakah yang ditampilkan, jika kode program pada file attachment dieksekusi?

```

1. class Student {
2.     private String id_number;
3.     private String name;
4.
5.     Student(String name){
6.         this.name = name;
7.     }
8.
9.     String getStudentData(){
10.        return "ID NUMBER : " + this.id_number + "-NAME:" + name;
11.    }
12.
13.    void setIdNumber(int id_number){
14.        this.id_number = name.charAt(0) + "" + id_number;
15.        System.out.println(getStudentData());
16.    }
17. }
18.
19. public class InformatikaStudent extends Student{
20.
21.     InformatikaStudent(String name){
22.         super(name);
23.     }
24.
25.     public static void main(String[] args) {
26.         InformatikaStudent student = new InformatikaStudent("Santy");
27.         student.setIdNumber(1102020);
28.     }
29. }

```

- ID NUMBER :-NAME:Santy
  - ID NUMBER :S1102020-NAME:
  - ID NUMBER :01102020-NAME:Santy
  - ID NUMBER :1102020-NAME:Santy**
  - ID NUMBER :S1102020-NAME:Santy
15. Pada saat membuat variabel pada interface, maka
- Tidak boleh menggunakan modifier public
  - Harus dengan inisialisasi value**
  - Harus menggunakan modifier private
  - Harus menggunakan keyword interface
  - Tidak boleh ada inisialisasi value
16. Manakah pernyataan berikut ini yang paling tepat tentang abstract?
- Abstract diturunkan menggunakan keyword implements
  - Abstract class tidak wajib menggunakan keyword abstract
  - Abstract class memperbolehkan terdapat method tanpa keyword abstract dengan implementasi program**
  - Abstract class tidak memperbolehkan deklarasi variabel tanpa nilai
  - Abstract class harus menggunakan variabel dengan modifier private dan protected
17. Perbedaan antara abstract dan interface adalah
- abstract tidak boleh menggunakan private, interface boleh
  - abstract semua method tidak boleh mempunyai isi program, interface boleh
  - penggunaan abstract memakai keyword implements, interface memakai keyword extends
  - interface tidak boleh menggunakan keyword protected, abstract boleh**
  - Tidak ada perbedaan antara abstract dan interface
18. Perhatikan kode program pada file attachment. Jika kode program tersebut dieksekusi maka akan terjadi error pada 3 baris. Baris berapa saja yang menjadi penyebab error?

```

1. interface Person{
2.     String name;
3.     public String getStudentData();
4.     public void setIdNumber(int id_number);
5. }
6.
7. class Student implements Person {
8.     private String id_number;
9.     private String name;
10.
11.     Student(String name){
12.         this.name = name;
13.     }
14.
15.     @Override
16.     public String getStudentData() {
17.         return ("ID NUMBER : " + this.id_number + "-NAME:" + name);
18.     }
19.
20.     @Override
21.     public void setIdNumber(int id_number) {
22.         this.id_number = name.charAt(0) + "" + id_number;
23.         return this.id_number;
24.     }
25. }
26.
27. public class InformatikaStudent extends Student{
28.
29.     InformatikaStudent(String name){
30.         super(name);
31.     }
32.
33.     public static void main(String[] args) {
34.         InformatikaStudent student = new InformatikaStudent("Santy");
35.         student.setIdNumber("1102020");
36.     }
37. }

```

- a. **2, 23 dan 17**
- b. 17, 23 dan 30
- c. 2, 34 dan 35
- d. 2, 23 dan 35
- e. 17, 34 dan 35

19. Nested Class terbagi menjadi 2 salah satunya adalah:

- a. **Static nested class**
- b. Non static integer nested class
- c. Static integer nested class
- d. Non static variable class
- e. Object class

```

1. class Kendaran Bermotor {
2.
3.     void infoKendaraan(){
4.         class SepedaMotor{
5.             int price;
6.             void infoMotor(){
7.                 System.out.println("Harga Motor: " + getPrice());
8.             }
9.
10.            void setMotor(int price){
11.                this.price = price;
12.            }
13.
14.            int getPrice(){
15.                return price;
16.            }
17.        }
18.
19.        SepedaMotor motorku = new SepedaMotor();
20.        motorku.setMotor(25000000);
21.        motorku.getPrice();
22.        motorku.infoMotor();
23.    }
24.
25.
26.
27.
28.
29.
30.
31.
32.

```

20. Kode program pada file attachment adalah contoh inner class dengan pendefinisian

- a. Anonymous static class
- b. Static nested class
- c. Anonymous inner class
- d. **Method local**
- e. Inner class