U paketu hr.fer.oopj definiran je razred R kako slijedi:

```
public class R {
  public double x;
  public double y;

  private R() {
  }

  public R(double x, double y) {
    this.x = x;
    this.y = y;
  }
}
```

U metodi main smještenoj u razredu X u istom paketu želimo deklarirati lokalnu varijablu r i pridružiti joj jedan novostvoreni objekt razreda R. Koji će od ponuđenih izraza to učiniti?

a
```
R r = new R;
```

b
```
R r = new R(3.14, 6.28);
```

c
```
R r = malloc(sizeof(R));
```

d
```
R r = new R();
```

Prilikom nadjačavanja neke metode povratni tip može biti:

a tip definiran u toj metodi

b sve od navedenog

c izvedena klasa iz tipa koji je definiran u toj metodi

d bilo koji tip

Koliko je objekata, primjeraka klase B spremno za uništavanje (garbage collection) u trenutku nakon što se izvrši naredba napisana u liniji 9?

```
public class B {
    B next;
    static B head;
    public static void main(String[] args) {
        B b = new B();
        b.next = new B();
        B.head = new B();
        B pom = B.head;
        B.head = b; //Line 9
        System.out.println("End");
    }
}
```

a 3

b 0

c 2

d 1

S kojim od navedenih načina se može definirati apstraktna metode method1?

a
```
abstract method1();
```

b
```
public virtual void method1();
```

c
```
abstract void method1(){};
```

d
```
public abstract void method1();
```

e
```
public abstract void method1;
```

Što će biti rezultat prevođenja i pokretanja programa?

```
interface A {
    int m();
}
class B implements A {
    public int m() {
        return 1;
    }
}
class C extends B {
    public int m() {
        return 2;
    }
}
public class Main {
    public static void main(String[] args) {
        A ref1 = new C();
        B ref2 = (B) ref1;
        System.out.println(ref2.m());
    }
}
```

a. Program se neće uspješno prevesti.

b. Program će se prevesti, ali neće ispisati ništa i/ili će se srušiti tijekom izvođenja.

c. Program će se prevesti i ispisat će `1` tijekom izvođenja.

d. Program će se prevesti i ispisat će `1 2` tijekom izvođenja.

e. Program će se prevesti i ispisat će `2` tijekom izvođenja.

U paketu hr.fer.oopj definiran je razred R kako slijedi:

```
public class R {
  public int x;
  public int y;

  public R(int x, int y) {
    this.x = x;
    this.y = y;
  }

  public String toString() { return x+","+y; }
}
```

Potom je u metodi main nekog drugog razreda napisano:

```
R[] r = new R[3];
for(int i = 0; i < r.length; i++) {
  r[i] = i==0 ? new R(1,1) : r[i-1];
}
r[1].x++;
System.out.println(r[0]);
System.out.println(r[1]);
```

```
System.out.println(r[0]);
System.out.println(r[1]);
```

Što će biti rezultat pokretanja programa?

a ispis 0.0 pa ispis 2.1

b program se zbog pogrešaka u razredu R neće prevesti

c ispis 1.0 pa ispis 1.0

d program se zbog pogrešaka u metodi main neće prevesti

Imamo klasu/razred Point. Koje metode možemo nadjačati iz klase Object?

a metodu equals

b metodu main

c metodu Point

d metodu toString

e metodu toArray

a u Javi nije moguće višestuko nasljeđivanje

b višestruko nasljeđivanje je kada neka klasa može naslijediti više klasa

c u Javi je moguće višestruko nasljeđivanje

d višestruko nasljeđivanje je kada klasa naslijedi klasu koja je naslijedila neku drugu klasu

e višestruko nasljeđivanje je klasa implementira više sučelja

Što od navedenog vrijedi za apstraktnu klasu

a ništa od navedenog

b mora imati barem jednu apstraktnu metodu

c s konstruktorom se može kreirati primjerak (instanca) ove klase

d primjerak potklase se može castati na tip ove klase

Razred R nasljeđuje razred P te oba imaju javni pretpostavljeni konstruktor. Koje od navedenih tvrdnji su točne?

a Naredba R r = new P(); će se prevesti bez greške.

b Na svaki primjerak razreda R može se gledati kroz referencu tipa P

c Prevoditelj će prilikom prevođenja naredbe P p = new R(); prijaviti grešku.

d Referenci tipa P ne možemo pridijeliti vrijednost pohranjenu u referencu tipa R bez da radimo eksplicitno ukalupljivanje pisanjem (P) ispred reference tipa R

e Referenci tipa P možemo pridijeliti vrijednost pohranjenu u referencu tipa R bez da radimo eksplicitno ukalupljivanje pisanjem (P) ispred reference tipa R

Razmotrite sljedeći kod i označite točne tvrdnje.

```
public class P {
    private int a;
    private P() {}
    public P(int a) {
        this.a = a;
    }
}
```

```
public class R extends P {
    private int b;
    public R() {}
    public R(int a) {
        this(a,5);
    }
    public R(int a,int b) {
        this.b=b;
        super(a);
    }
}
```

a Pretpostavljeni konstruktor razreda R je neispravan jer u njemu nismo napisali super();

b Pretpostavljeni konstruktor razreda R je neispravan jer pokušava pozvati nadređeni konstruktor koji mu nije vidljiv

c Konstruktor R(int a, int b) je neispravan jer koristi super, a ako se on koristi, onda mora biti prva naredba u tijelu konstruktora

d Razred P ne može imati dva konstruktora

e Konstruktor R(int a) je neispravan jer je u njemu umjesto this(a,5) trebalo stajati super(a,5)

Od ponuđenih tipova odaberite sve tipove koji se mogu pridijeliti varijabli referenci *ref*, kako bi se program uspješno preveo?

```
interface Perishable { }
abstract class Food implements Perishable { }
class IceCream extends Food { }
public class Main {
    public static void main(String[] args) {
        IceCream [] refArray = new IceCream[] { new IceCream()};
        for(Food f : refArray) {
            _____ ref = f;
        }
    }
}
```

a Food

b Main

c Object

d Perishable

e IceCream

Od ponuđenih tipova odaberite sve tipove koji se mogu pridijeliti varijabli referenci *ref*, kako bi se program uspješno preveo?

```
interface Perishable { }
abstract class Food implements Perishable { }
class IceCream extends Food { }
public class Main {
    public static void main(String[] args) {
        IceCream [] refArray = new IceCream[] { new IceCream()};
        for(Food f : refArray) {
            _____ ref = f;
        }
    }
}
```

a Food

b Main

c Object

d Perishable

e IceCream

Koliko je objekata, primjeraka klase B spremno za uništavanje (garbage collection) u trenutku nakon što se izvrši naredba napisana u liniji 8?

```
public class B {
    B next;
    static B head;
    public static void main(String[] args) {
        B b = new B();
        b.next = new B();
        B.head = new B();
        b = B.head; //Line 8
        System.out.println("End");
    }
}
```

a 0

b 2

c 3

d 1

Što ispisuje sljedeći program?

```
package hr.fer.oop.lab2.example;
public class StrangeSum {
    public final int Ssum1(int x, int y) {
        return x + y + 10;
    }
    public int Ssum2(int x, int y) {
        return x + y + 10;
    }
}
```

```
package hr.fer.oop.lab2.example;
public class TwoNumbers extends StrangeSum{
    @Override
    public int Ssum1(int x, int y) {
        return x + y + 15;
    }
    public static void main(String[] args) {
        TwoNumbers ss = new TwoNumbers();
        int result = ss.Ssum1(1, 1);
        System.out.println(result);
    }
}
```

a Ništa od navedenog

b 2

c 17

d 12

U paketu hr.fer.oopj definiran je razred R kako slijedi:

```
public class R {
  public double x;
  public double y;

  public R() {
    this(3,5);
  }

  public R(int x) {
    y = 3;
    this(x,x);
  }

  public R(int x, int y) {
    this((double)x,(double)y);
  }

  public R(double x, double y) {
    this.x = x;
    this.y = y;
  }
}
```

Što će biti rezultat prevođenja napisanog izvornog koda u byte-kod?

a. Prevodilac će prijaviti pogrešku kod konstruktora R(int x)

b. Prevodilac će prijaviti pogrešku da ne mogu postojati dva konstruktora koja oba primaju dva argumenta

c. Prevodilac će prijaviti pogrešku kod konstruktora R(int x, int y)

d. Kod će se uredno prevesti i nastat će class-datoteka

Razmotrite sljedeći kod i označite točne tvrdnje.

```
public class P {
    private int a;
    private P() {}
    public P(int a) {
        this.a = a;
    }
}
```

```
public class R extends P {
    private int b;
    public R() {}
    public R(int a) {
        this(a,5);
    }
    public R(int a,int b) {
        this.b=b;
        super(a);
    }
}
```

a Konstruktor R(int a, int b) je neispravan jer koristi super, a ako se on koristi, onda mora biti prva naredba u tijelu konstruktora

b Pretpostavljeni konstruktor razreda R je neispravan jer u njemu nismo napisali super();

c Konstruktor R(int a) je neispravan jer je u njemu umjesto this(a,5) trebalo stajati super(a,5)

d Razred P ne može imati dva konstruktora

e Pretpostavljeni konstruktor razreda R je neispravan jer pokušava pozvati nadređeni konstruktor koji mu nije vidljiv

Razred R nasljeđuje razred P te oba imaju javni pretpostavljeni konstruktor. Koje od navedenih tvrdnji točne?

a Naredba R r = new P(); će se prevesti bez greške.

b Naredba P p = new R(); će se prevesti bez greške.

c Prevodilac će prilikom prevođenja naredbe R r = new P(); prijaviti grešku.

d Na svaki primjerak razreda P može se gledati kroz referencu tipa R

e Referenci tipa P ne možemo pridijeliti vrijednost pohranjenu u referencu tipa R bez da radimo eksplicitno ukalupljivanje pisanjem (P) ispred reference tipa R

Što od navedenog vrijedi za apstraktnu klasu

a mora imati barem jednu apstraktnu metodu

b s konstruktorom se može kreirati primjerak (instanca) ove klase

c primjerak potklase se može castati na tip ove klase

d ništa od navedenog

Što je od navedenog točno ako u programskom kodu piše

```
class A extends B
```

```
class C extends A
```

a Klasa A je specijalizacija klase B

b Klasa C je specijalizacija klase A

c Klasa B je generalizacija klase A

d Klasa C je generalizacija klase A

e Klasa B je specijalizacija klase A

U paketu hr.fer.oopj definiran je razred R kako slijedi:

```
public class R {
  public double x;
  public double y;

  public R() {
    ____________________
  }

  public R(double x, double y) {
    this.x = x;
    this.y = y;
  }
}
```

Što će od ponuđenoga, napisano na mjestu označenom podvlakama, osigurati da se pozivom pretpostavljenog konstruktora u konačnici obavi inicijalizacija objekta postavljanjem članske varijable x na 3 te članske varijable y na 5?

a `R(3,5);`

```
  public R(double x, double y) {
    this.x = x;
    this.y = y;
  }
}
```

Što će od ponuđenoga, napisano na mjestu označenom podvlakama, osigurati da se pozivom pretpostavljenog konstruktora u konačnici obavi inicijalizacija objekta postavljanjem članske varijable x na 3 te članske varijable y na 5?

a `R(3,5);`

b `this.3.5;`

c `this(3,5);`

d `this("3","5");`

U paketu hr.fer.oopj definiran je razred R kako slijedi:

```
public class R {
  public double x;
  public double y;

  public R() {
    this(3,5);
  }

  public R(int x) {
    y = 3;
    this(x,x);
  }

  public R(int x, int y) {
    this((double)x,(double)y);
  }

  public R(double x, double y) {
    this.x = x;
    this.y = y;
  }
}
```

Što će biti rezultat prevođenja napisanog izvornog koda u byte-kod?

a Prevodilac će prijaviti pogrešku da ne mogu postojati dva konstruktora koja oba primaju dva argumenta

b Prevodilac će prijaviti pogrešku kod konstruktora R(int x, int y)

c Kod će se uredno prevesti i nastat će class-datoteka

d Prevodilac će prijaviti pogrešku kod konstruktora R(int x)

U paketu hr.fer.oopj definiran je razred R kako slijedi:

```
public class R {
  public int x;
  public int y;

  public R(int x, int y) {
    this.x = x;
    this.y = y;
  }

  public String toString() { return x+","+y; }

  public boolean equals(Object o) {
    R other = (R)o;
    return this.x==other.x &amp;amp;&amp;amp; this.y==other.y;
  }

}
```

Potom je u metodi main nekog drugog razreda napisano:

```
R r1 = new R(2,3);
R r2 = new R(2,3);
R r3 = new R(3,2);
System.out.println(r1.equals(r2));
System.out.println(r1.equals(r3));
```

Što će biti rezultat pokretanja programa?

a ispis true pa ispis true

b ispis true pa ispis false

c program se zbog pogrešaka u metodi main neće prevesti

d ispis false pa ispis false

Prilikom nadjačavanja neke metode povratni tip može biti:

a sve od navedenog

b bilo koji tip

c izvedena klasa iz tipa koji je definiran u toj metodi

d tip definiran u toj metodi

Deklarirani su sljedeći tipovi:

```
interface I1 { }
interface I2 { }
class C1 { }
class C2 { }
```

Označite sve dozvoljene deklaracije.

a
```
interface I4 extends I1, I2 { }
```

b
```
class C4 extends C1 implements I1, I2 { }
```

c
```
class C3 implements I1, I2 { }
```

d
```
interface I5 implements I1 { }
```

e
```
interface I3 extends I1 { }
```

Razred P nasljeđuje razred R te oba imaju javni pretpostavljeni konstruktor. Koje od navedenih tvrdnji su točne?

a Referenci tipa R ne možemo pridijeliti vrijednost pohranjenu u referencu tipa P bez da radimo eksplicitno ukalupljivanje pisanjem (R) ispred reference tipa P

b Na svaki primjerak razreda R može se gledati kroz referencu tipa P

c Naredba P p = new R(); će se prevesti bez greške.

d Naredba R r = new P(); će se prevesti bez greške.

e Prevodilac će prilikom prevođenja naredbe P p = new R(); prijaviti grešku.