(A) CONOSCENZA TERMINOLOGICA

Dare una breve descrizione dei termini introdotti

- **toString()**
- **finalize()**
- **clone()**
- **hashCode()**
- **equals()**
- **getClass()**
- **Object**

(B) CONOSCENZA E COMPETENZA

Rispondere alle seguenti domande producendo anche qualche esempio

Conoscenza

1. Quali sono i metodi principali della classe **Object**?
2. Quale *gerarchia di classi* parte da **Object**?
3. Quale vantaggio presenta il fatto che la classe **Object** *sia superclasse di tutte le classi*?

Competenza

1. Cosa indica **Object**?
2. Qual è il formato tipico che descrive lo *stato di un oggetto*?
3. Quali sono i metodi tipici della classe **Object**?
4. Qual è la principale utilità del metodo **toString**()?
5. Come si può verificare che *due oggetti abbiano lo stesso contenuto*?
6. Come si può verificare l'identità di due oggetti, ossia che siano lo stesso oggetto?

(C) ESERCIZI DI COMPRENSIONE

1. Quando si dichiara una classe, l'uso della parola chiave ……. indica che la classe è derivata da un'altra classe. Se la parola …….. viene omessa, la nostra classe diventa derivata della classe ……., che rappresenta la superclasse di tutte le classi.
2. Il metodo **clone**() crea un oggetto ……. dell'oggetto su cui è istanziato e carica negli attributi valori …….. rispetto all'oggetto di partenza. I due oggetti ottenuti ……. la stessa cosa, ma non sono lo …….. oggetto.
3. Il metodo **toString**() restituisce una ……. che rappresenta l'oggetto. Il formato restituito è del tipo ……., dove ……. indica il nome della classe, mentre …….. indica un numero esadecimale che individua l'oggetto in ……. . Normalmente, questo formato viene convertito in un formato più leggibile del tipo *nomeclasse* = […… = ……., ….. = …., … …….. = …….].
4. Il metodo **hashCode**() restituisce un …… che individua in modo …….. l'oggetto su cui è ……. .
5. Il metodo **equals**() consente di verificare se due oggetti hanno lo stesso …….. .Per verificare se due oggetti sono lo stesso ……. si usa l'operatore …….. .
6. Il metodo **getClass**() è molto comodo perché restituisce il ……. della classe a cui appartiene l'oggetto su cui esso è ……
7. Il metodo **finalize**() viene invocato dal ..... …… quando un oggetto non viene più ……. .e provvede ad eliminarlo dalla memoria. In genere, il metodo viene ridefinito, in modo da fargli svolgere altre operazioni di …….
8. Data la dichiarazione:
   Integer n1 = **new Integer**(1), n2 = **new Integer**(1);
   cosa stampa la seguente istruzione?
   if (n1.**equals**(n2))
   System.out.println("Gli oggetti sono uguali");
   else
   System.out.println("Gli oggetti non sono uguali");
9. Stabilire cosa ritorna il seguente metodo:
   void f (**Object** obj)
   { **System.out.println**(obj.**getClass**());
   }
10. Data la classe *Punto*, con attributi *ascissa* e *ordinata* si deve ridefinire il metodo **toString**() per stampare un oggetto nella classica forma matematica *P(ascissa, ordinata)*. Completare la codifica del metodo suddetto:
    String ……. (**Object** obj)
    { **return** ……;
    }
11. Considerata la classe *Punto* con attributi x ed y, analizzare il seguente frammento di applicazione, descrivere il tipo di errore e in quale momento si verifica.
    import java.util.*;
    import java.awt.*;
    class oggetti
    { **public static void main**(**String** args[])
    { **Object** o;

```
        Point p=new Point(1,2);;
        System.out.println(o.x);
        }
}
```

12. Per ciascuno dei seguenti metodi della classe **Object**, descriverne la sintassi.

Metodo	Sintassi
Uguaglianza di due oggetti	
Codice identificativo dell'oggetto	
Nome della classe dell'oggetto	
Rappresentazione stringa dell'oggetto	
Copia dell'oggetto	

13. Per ciascuna delle seguenti frasi, indicare se vera o falsa.

	Vero	Falso
Istanze multiple di **hashCode**() sullo stesso oggetto durante l'esecuzione di un'applicazione, danno stesso valore		
Gli *hash code* degli oggetti non variano da un'esecuzione all'altra		
Se c1.**equals**(c2) = **true**, c1 e c2 hanno anche lo stesso *hash code*		
Gli operatori **equals**(…) e "==" indicano la stessa cosa		
Se si usa il metodo **equals**(…) sui tipi primitivi, si ha errore		
Se c1.**hashCode**() == c2.**hashCode**() allora *c1* e *c2* sono uguali		
Tutti gli oggetti si possono memorizzare in una variabile **Object**		

14. Per ciascuno dei seguenti metodi di Object, scrivere la corretta firma

	Firma
toString()	
getClass()	
equals()	
clone()	
finalize()	

(D) ESERCIZI DI APPLICAZIONE

1. Scrivere una semplice applicazione che permetta di verificare che il metodo **equals**() implementa una relazione di equivalenza, ossia gode delle proprietà:
   a. riflessiva: un oggetto *x* è uguale a se stesso;
   b. simmetica: se l'oggetto *x* è uguale all'oggetto *y*, anche *y* risulterà uguale ad *x*;
   c. transitiva: se l'oggetto *x* è uguale all'oggetto *y* e l'oggetto *y* è uguale all'oggetto *z* , anche *x* risulterà uguale ad *z*;
2. Scrivere un'applicazione che crei una classe *Frutta*, con due attributi: *nome*, di tipo stringa, e *colore* di classe **Color**. Nella classe, ridefinire il metodo **equals**() in modo che, se l'argomento:
   - è **null**, restituisca **false**;
   - è di classe *Frutta* (verificabile con l'operatore **instanceof**), istanzia **equals**() sull'attributo *nome*;
   - non è di classe *Frutta*, restituisca **false**.

   Successivamente, creare due oggetti di classe *Frutta*, ad esempio *mela* e *arancia* e verificare che:
   a. il metodo **equals**() su essi dà **false**;
   b. il metodo **hashCode**() su ciascuno, dà valori diversi.
3. Scrivere un metodo che, nella classe *Punto* consenta la stampa di un punto p nella forma consueta *p(ascissa, ordinata)*.
4. Aggiungere alla classe *Rettangolo* la ridefinizione del metodo **toString**() in modo che stampi la descrizione testuale di un oggetto nella forma: Rettangolo [base=…, altezza=…]
5. Aggiungere alla classe *Quadrato* la ridefinizione del metodo **toString**() in modo che stampi la descrizione testuale di un oggetto nella forma: Quadrato [lato=…]
6. Aggiungere alla classe *Angolo* la ridefinizione del metodo **toString**() in modo che stampi la descrizione testuale di un oggetto nella forma: Angolo [gradi=…, primi=…, secondi=…]

Completare le seguenti proposizioni

Associare le proposizioni di sinistra con le corrispondenti sulla destra:
Completare le seguenti tabelle:
Individuare le proposizioni vere/false
Esercizi pratici

La numerazione è progressiva atttraverso le varie tipologie di esercizi

Completare le seguenti proposizioni

1. Una classe di problemi è formata da tutti i problemi aventi ……...

Associare le proposizioni di sinistra con le corrispondenti sulla destra:

1	L'analisi del testo…	**A**	elencare gli input e gli output
2	La tabella delle variabili di I/O…	**B**	descrivere le specifiche del problema
3	Il modello del problema…	**C**	descrivere sinteticamente la soluzione
4	Il procedimento risolutivo…	**D**	rappresentare il tipo di problema

Completare le seguenti tabelle:

IDClasse	Classe	Sezione	Specializzazione
11	3	A	Informatica
12	4	A	Informatica
13	5	A	Informatica
14	3	B	Elettronica
15	4	B	Elettronica
16	5	B	Elettronica
17	3	C	**NULL**

Domande vero/falso:

	Vero	Falso

Esercizi pratici

(E) ESERCITAZIONI PRATICHE