

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO | 2° ANO EICO013 | ALGORITMOS E ESTRUTURAS DE DADOS | 2013-2014 - 1° SEMESTRE

Parte teórica. Duração: 30m

| Nome: | _Código: |
|--|----------|
| | |
| Notas: - Responda às questões seguintes, indicando a opção correta (em maiúsculas) | |

1. Considere os dois membros-função calcula1 e calcula2 de uma classe:

- Cada resposta errada vale -20% da cotação da pergunta

```
class ABC {
public:
   static unsigned int calcula1(...);
   int calcula2(...);
};
```

- A. Na função calcula2 não se pode chamar a função calcula1 porque calcula2 é static e calcula1 não
- B. Na função *calcula*2 não se pode chamar a função *calcula*1 porque *calcula*2 tem um tipo de retorno diferente do tipo de retorno de *calcula*1
- C. Pelo facto de haver um membro-função static a classe é abstrata.
- D. Durante a execução do programa, existe uma implementação de *calcula1* em memória mesmo que não existam objetos dessa classe.
- E. Nenhuma das possibilidades anteriores

| Resposta: | |
|-----------|--|
| | |

para fazer overloading da função soma. É verdade que as declarações:

- A. Estão erradas porque a segunda declaração tem mais um argumento que as outras
- B. São válidas e representam um exemplo correto de overloading de funções
- C. Estão erradas porque no *overloading* de funções é obrigatório as declarações terem o mesmo nome só se podendo variar o número de argumentos e não o seu tipo
- D. Estão erradas porque no *overloading* de funções é obrigatório as declarações terem o mesmo nome só se podendo variar o tipo dos argumentos e não o seu número
- E. Nenhuma das possibilidades anteriores

| Res | posta: | |
|-----|--------|--|
| | | |

- 3. Numa hierarquia de classes C++ é verdade que se houver um membro-função virtual puro na classe base:
 - A. Obrigatoriamente todas as classes derivadas são abstratas
 - B. Uma classe derivada que não implemente essa função é abstrata
 - C. O programa só pode criar um objeto dessa classe base
 - D. A implementação desse membro-função tem que ser feita num ficheiro separado
 - E. Nenhuma das possibilidades anteriores

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO | 2º ANO EICO013 | *ALGORITMOS E ESTRUTURAS DE DADOS* | 2013-2014 - 1º SEMESTRE

4. Considere a seguinte definição da classe Classe 1:

```
class Classe1 {
private:
  int info;
public:
  Classe1(int info);
}:
```

- A. A declaração Classe1 c; dá erro de compilação
- B. A declaração *Classe1 c*; não dá erro de compilação, pois o C++ fornece um construtor sem argumentos por omissão
- C. A definição da classe não compila corretamente porque não tem um membro-função para acesso ao membro-dado
- D. A declaração *Classe1 c*; dá erro de compilação porque *info* não pode ser usado em simultâneo como membro-dado e como parâmetro do construtor
- E. Nenhuma das possibilidades anteriores

5. A classe **Mestrando** é uma classe derivada da classe **Estudante**, que é uma classe derivada da classe **Pessoa**. Considere as seguintes declarações de variáveis:

Pessoa *p=new Pessoa(); Estudante *e=new Estudante(); Mestrando *m =new Mestrando(); Quais das seguintes atribuições estão corretas?

| I. $p = m;$ | <pre>II. p = new Mestrando();</pre> | III. $m = new Estudante();$ |
|---------------|-------------------------------------|-----------------------------|
| IV. $m = p$; | <pre>v. e = new Pessoa();</pre> | |

- A. III e IV
- B. lelV
- C. lell
- D. II, III e V
- E. Nenhuma das possibilidades anteriores

- 6. Qual das seguintes afirmações é verdadeira, relativa a um membro-função abstrato herdado por ClasseX?
 - A. Tem de ser obrigatoriamente definido em ClasseX para que ClasseX possa ser instanciada
 - B. Não permite que ClasseX seja uma classe base
 - C. Obriga a que ClasseX seja sempre uma classe abstrata
 - D. Redefine qualquer membro-função em ClasseX com o mesmo nome
 - E. Nenhuma das possibilidades anteriores

| Res | posta: | : |
|-----|--------|---|
| | posta. | • |



MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO | 2° ANO EICO013 | *ALGORITMOS E ESTRUTURAS DE DADOS* | 2013-2014 - 1° SEMESTRE

| Noi | me: | | | | | | Código: | |
|-----|----------------|---|--|--|---|----------------------------------|----------------|---------------------|
| | | | | | | | | |
| 7. | Con | isidere as seguir | ites declaraçõe | s: | | | | |
| | | class A; | class B; | class C: p | oublic A; | class D: p | oublic A, p | ublic B; |
| | B. C. D. | A declaração o A declaração o | da classe D só so da classe D só so ooderá impleme | erá possível se a erá possível se a entar métodos es | , portanto a dec classe A for pura classe B for pura státicos, herdado | amente abstrac amente abstrac | ta | ada |
| | Res | posta: | | | | | | |
| 8. | | | | | o) implementam re cada uma das | | ring print | () , que |
| | | class A; | class B; | class C: p | oublic A; | class D: p | oublic A, p | ublic B; |
| | A. | O método pri | | entado em D , é | um polimorfismo | o do mesmo m | étodo, herdad | lo de A e B |
| | В. | Se o método p | orint() for de | | <i>irtual</i> em A , ent | | lasses polimór | ficas de A , |
| | _ | por reimplementarem diferentes comportamentos para print () | | | | | | |
| | C. | As classes A e B são consideradas polimórficas entre si, pois implementam diferentes comportamentos para o mesmo método print() | | | | | | |
| | | • | | | olimórficas entre | e si | | |
| | E. | Nenhuma das _I | possibilidades a | interiores | | | | |
| | Res | posta: | | | | | | |
| 9. | | al das seguintes | | | | | | |
| | | | | m de ser obrigat to pode ser cons | oriamente inicia | lizado no const | rutor | |
| | | | | • | der a um membr | o-dado estático | 0 | |
| | | | • | • | der a membros-d | ado não estátio | cos | |
| | Ł. | Nenhuma das ¡ | possibilidades a | interiores | | | | |
| | Res | posta: | | | | | | |
| 10. | Con | nsidere as seguir | ntes declaraçõe | s: | | | | |
| | | class | Base; | class Deri | vada: protec | ted Base; | | |
| | | | • | | s protegidos da c | | | |
| | | | | | a ser protegidos | | | |
| | | | | | a ser protegidos penas acessível p | | | |
| | | Nenhuma das į | • | | | | | |
| | Res | posta: | | | | | | |