**A+ Computer Science**

**Methods M/C TEST**

**Directions ::** On your answer sheet, mark the letter of the best answer to each question.

1. **What is a method?**

|  |  |
| --- | --- |
| a. | a tool that helps organize a program |
| b. | a tool that complicates a program |
| c. | a tool that makes a program longer |
| d. | a tool that makes a program harder to read |
| e. | a tool that makes a program more disorganized |

2. **When writing a method, you must always have an open and close what ?**

|  |  |
| --- | --- |
| a. | parenthesis |
| b. | brace |
| c. | bracket |
| d. | A and B only |
| e. | A, B, and C |

3. **Given the following statement, var must be defined as which of the following types?**

var=keyboard.nextInt();

|  |  |
| --- | --- |
| a. | short |
| b. | byte |
| c. | int |
| d. | char |
| e. | String |

4. **Given the following statement, var must be defined as which of the following types?**

var=keyboard.nextLong();

|  |  |
| --- | --- |
| a. | short |
| b. | byte |
| c. | int |
| d. | long |
| e. | String |

5. **Given the following statement, var must be defined as which of the following types?**

var = keyboard.nextDouble();

|  |  |
| --- | --- |
| a. | short |
| b. | double |
| c. | int |
| d. | long |
| e. | String |

6. **Given the following statement, var must be defined as which of the following types?**

var = keyboard.nextShort();

|  |  |
| --- | --- |
| a. | short |
| b. | double |
| c. | byte |
| d. | char |
| e. | String |

7. **Given the following statement, var must be defined as which of the following types?**

var = keyboard.nextFloat();

|  |  |
| --- | --- |
| a. | float |
| b. | byte |
| c. | char |
| d. | boolean |
| e. | String |

8. **Given the following statement, var must be defined as which of the following types?**

var = keyboard.nextByte();

|  |  |
| --- | --- |
| a. | bytes |
| b. | byte |
| c. | bite |
| d. | boolean |
| e. | String |

9. **Given the following statement, var must be defined as which of the following types?**

var = keyboard.nextLine();

|  |  |
| --- | --- |
| a. | float |
| b. | byte |
| c. | char |
| d. | boolean |
| e. | String |

10. **Given the following statement, var must be defined as which of the following types?**

var = keyboard.next();

|  |  |
| --- | --- |
| a. | float |
| b. | byte |
| c. | char |
| d. | boolean |
| e. | String |

11. **Which of the following Scanner methods can be used to read in the value** 45 **?**

|  |  |
| --- | --- |
| a. | nextByte() |
| b. | nextShort() |
| c. | nextInt() |
| d. | nextDouble() |
| e. | all of these |

12. **Which of the following Scanner methods can be used to read in the value** 32760 **?**

|  |  |
| --- | --- |
| a. | nextShort() |
| b. | nextInt() |
| c. | nextByte() |
| d. | A and B only |
| e. | all of these |

13. **Which of the following Scanner methods can be used to read in the value** 65535 **?**

|  |  |
| --- | --- |
| a. | nextShort() |
| b. | nextInt() |
| c. | nextByte() |
| d. | A and B only |
| e. | all of these |

14. **Which of the following Scanner methods can be used to read in the value**  52.245  **?**

|  |  |
| --- | --- |
| a. | nextDouble() |
| b. | nextLong() |
| c. | nextByte() |
| d. | nextInt() |
| e. | A and B only |

15. **Which of the following Scanner methods can be used to read in the value** get it **?**

|  |  |
| --- | --- |
| a. | next() |
| b. | nextLine() |
| c. | nextByte() |
| d. | nextInt() |
| e. | A and B only |

16. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

}

//code in the main of another class

CS test = new CS();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | oneone |
| c. | oneoneone |
| d. | oneoneoneone |
| e. | Oneoneoneoneone |

17. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

}

//code in the main of another class

CS test = new CS();

test.one();

test.one();

test.one();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | oneone |
| c. | oneoneone |
| d. | oneoneoneone |
| e. | oneoneoneoneone |

18. **What is output by the code below?**

public class CS{

public void one(){

System.out.print("one");

}

}

//code in the main of another class

CS test = new CS();

test.one();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | oneone |
| c. | oneoneone |
| d. | oneoneoneone |
| e. | oneoneoneoneone |

19. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

}

//code in the main of another class

CS test = new CS();

test.one();

test.one();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | oneone |
| c. | oneoneone |
| d. | oneoneoneone |
| e. | oneoneoneoneone |

20. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

}

//code in the main of another class

CS test = new CS();

test.one();

test.one();

test.one();

test.one();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | oneone |
| c. | oneoneone |
| d. | oneoneoneone |
| e. | oneoneoneoneone |

21. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

public void two()

{

System.out.print("two");

}

}

//code in the main of another class

CS test = new CS();

test.one();

test.two();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | onetwo |
| c. | onetwoone |
| d. | onetwoonetwo |
| e. | onetwoonetwoone |

22. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

public void two()

{

System.out.print("two");

}

}

//code in the main of another class

CS test = new CS();

test.two();

test.one();

test.two();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | onetwo |
| c. | twoonetwoone |
| d. | onetwoonetwo |
| e. | onetwoonetwoone |

23. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

two();

}

public void two()

{

System.out.print("two");

}

}

//code in the main of another class

CS test = new CS();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | onetwo |
| c. | onetwoone |
| d. | onetwoonetwo |
| e. | onetwooneoneone |

24. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

public void two()

{

System.out.print("two");

one();

}

}

//code in the main of another class

CS test = new CS();

test.one();

test.two();

test.two();

|  |  |
| --- | --- |
| a. | one |
| b. | onetwo |
| c. | onetwoone |
| d. | onetwoonetwo |
| e. | onetwoonetwoone |

25. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

}

public void two()

{

System.out.print("two");

one();

}

}

//code in the main of another class

CS test = new CS();

test.one();

test.two();

test.one();

|  |  |
| --- | --- |
| a. | onetwooneone |
| b. | oneoneoneone |
| c. | twotwotwotwo |
| d. | onetwoonetwo |
| e. | twooneoneone |

26. **Consider the class and statements that follow.**

public class Dog

{

public void barkOnce()

{

System.out.print("woof");

}

public void barkTwice()

{

barkOnce();

barkOnce();

}

}

Given the declaration below, which of the following are valid statements?

Dog d = new Dog();

I.

d.barkOnce();

II.

d.barkTwice();

III.

System.out.println( d.barkTwice() );

|  |  |
| --- | --- |
| a. | I only |
| b. | II only |
| c. | III only |
| d. | I and II only |
| e. | II and III only |

27. **Consider the class and statements that follow.**

public class Dog

{

private void barkOnce()

{

System.out.print("woof");

}

public void barkTwice()

{

barkOnce();

barkOnce();

}

}

Given the declaration below, which of the following are valid statements?

Dog d = new Dog();

I.

d.barkOnce();

II.

d.barkTwice();

III.

System.out.println( d.barkTwice() );

|  |  |
| --- | --- |
| a. | I only |
| b. | II only |
| c. | III only |
| d. | I and II only |
| e. | none of the these statements are valid |

28. **Consider the class and statements that follow.**

public class Dog

{

private String voice;

public void setVoice( String v )

{

voice = v;

}

public void barkOnce()

{

System.out.print( voice );

}

public void barkTwice()

{

barkOnce();

barkOnce();

}

}

Given the declaration below, which of the following are valid statements?

Dog d = new Dog();

d.setVoice( “roof” );

I.

d.barkOnce();

II.

d.barkTwice(“rrrrrr”);

III.

System.out.println( d.barkTwice() );

|  |  |
| --- | --- |
| a. | I only |
| b. | II only |
| c. | III only |
| d. | I and II only |
| e. | none of the these statements are valid |

29. **Consider the class and statements that follow.**

public class Dog

{

private String voice;

public void setVoice( String v )

{

voice = v;

}

public void barkOnce()

{

System.out.print( voice );

}

public void barkTwice()

{

barkOnce();

barkOnce();

}

}

Given the declaration below, which of the following are valid statements?

Dog d = new Dog();

d.setVoice( “roof” );

I.

d.barkOnce();

II.

d.barkTwice();

III.

System.out.println( d.barkTwice() );

|  |  |
| --- | --- |
| a. | I only |
| b. | II only |
| c. | III only |
| d. | I and II only |
| e. | none of the these statements are valid |

30. **Consider the class and statements that follow.**

public class Dog

{

private String voice;

public void setVoice( String v )

{

voice = v;

}

public void barkOnce()

{

System.out.print( voice );

}

public void barkTwice()

{

barkOnce();

barkOnce();

}

}

Given the declaration below, which of the following are valid statements?

Object d = new Dog();

I.

d.barkOnce();

II.

d.barkTwice(“rrrrrr”);

III.

System.out.println( d.barkTwice() );

|  |  |
| --- | --- |
| a. | I only |
| b. | II only |
| c. | III only |
| d. | I and II only |
| e. | none of the these statements are valid |

31. **Consider the class and statements that follow.**

public class Dog

{

private String voice;

public void setVoice( String v )

{

voice = v;

}

private void barkOnce()

{

System.out.print( voice );

}

public void barkTwice()

{

barkOnce();

barkOnce();

}

public String toString()

{

return voice;

}

}

Given the declaration below, which of the following are valid statements?

Object d = new Dog();

((Dog)d).setVoice( “roof” );

I.

d.barkOnce();

II.

((Dog)d).barkTwice();

III.

System.out.println( d );

|  |  |
| --- | --- |
| a. | I only |
| b. | II only |
| c. | I and III only |
| d. | II and III only |
| e. | none of the these statements are valid |

32. **Consider the class and statements that follow.**

public class Money

{

private double amount;

public Money(double am)

{

amount = am;

}

public void addQuarters(int q)

{

amount += q \* 0.25;

}

public void addDimes(int d)

{

amount += d \* 0.1;

}

public double getAmount()

{

return amount;

}

}

What is the output of the code below?

Money m = new Money(.3);

m.addQuarters(3);

m.addDimes(6);

m.addQuarters(4);

System.out.println(m.getAmount());

|  |  |
| --- | --- |
| a. | 2.65 |
| b. | 2.45 |
| c. | 13.3 |
| d. | 2.35 |
| e. | 3.25 |

33. **Consider the class and statements that follow.**

public class Money

{

private double amount;

public Money(double am)

{

amount = am;

}

public void addQuarters(int q)

{

amount += q \* 0.25;

}

public void addDimes(int d)

{

amount += d \* 0.1;

}

public double getAmount()

{

return amount;

}

}

What is the output of the code below?

Money m = new Money(2.1);

m.addDimes(7);

m.addQuarters(7);

m.addDimes(2);

System.out.println(m.getAmount());

|  |  |
| --- | --- |
| a. | 2.1 |
| b. | 2.65 |
| c. | 5.05 |
| d. | 4.75 |
| e. | 18.1 |

34. **Consider the code that follows.**

Scanner scan = new Scanner("4 3 2 5 1");

int sum = 0;

while(scan.hasNextInt())

{

/\* code \*/

}

Which of the following are valid statements?

I.

System.out.println(scan.next());

II.

sum += scan.next();

III.

System.out.println(scan.nextInt());

|  |  |
| --- | --- |
| a. | I only |
| b. | II only |
| c. | I and III only |
| d. | II and III only |
| e. | none of the these statements are valid |

35. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

two(true);

}

public void two(boolean stop)

{

System.out.print("two");

if(!stop)

one();

}

}

///////////////////////////////////////////////////////////

//code in the main of another class

CS test = new CS();

test.two(false);

|  |  |
| --- | --- |
| a. | two |
| b. | twoone |
| c. | twoonetwo |
| d. | twoonetwoone |
| e. | onetwooneoneone |

36. **What is output by the code below?**

public class CS

{

public void one()

{

System.out.print("one");

two(true);

}

public void two(boolean stop)

{

System.out.print("two");

if(!stop)

one();

}

}

///////////////////////////////////////////////////////////

//code in the main of another class

CS test = new CS();

test.one();

|  |  |
| --- | --- |
| a. | one |
| b. | onetwo |
| c. | onetwoone |
| d. | onetwoonetwo |
| e. | onetwooneoneone |

37. **What is output by the code below?**

public class CS

{

int q = 0;

public void go(int i)

{

if(i==0)

q++;

for(int j = 0;j<i;j++)

go2(i-1);

}

public void go2(int i)

{

if(i==0)

q++;

for(int j = 0;j<i;j++)

go(i-1);

}

public int getQ()

{

return q;

}

}

///////////////////////////////////////////////////////////

//code in the main of another class

CS test = new CS();

test.go(2);

System.out.println( test.getQ() );

|  |  |
| --- | --- |
| a. | 1 |
| b. | 2 |
| c. | 3 |
| d. | 4 |
| e. | 5 |

38. **What is output by the code below?**

public class CS

{

int q = 0;

public void go(int i)

{

if(i==0)

q++;

for(int j = 0;j<i;j++)

go2(i-1);

}

public void go2(int i)

{

if(i==0)

q++;

for(int j = 0;j<i;j++)

go(i-1);

}

public int getQ()

{

return q;

}

}

///////////////////////////////////////////////////////////

//code in the main of another class

CS test = new CS();

test.go(3);

System.out.println( test.getQ() );

|  |  |
| --- | --- |
| a. | 4 |
| b. | 6 |
| c. | 8 |
| d. | 10 |
| e. | 18 |

39. **What is output by the code below?**

public class CS

{

int q = 0;

public void go(int i)

{

if(i==0)

q++;

for(int j = 0;j<i;j++)

go2(i-1);

}

public void go2(int i)

{

if(i==0)

q++;

for(int j = 0;j<i;j++)

go(i-1);

}

public int getQ()

{

return q;

}

}

//code in the main of another class

CS test = new CS();

test.go(4);

System.out.println( test.getQ() );

|  |  |
| --- | --- |
| a. | 12 |
| b. | 16 |
| c. | 18 |
| d. | 24 |
| e. | 64 |

40. **How is nextLine() different from other scanner methods?**

|  |  |
| --- | --- |
| a. | It will never throw an exception |
| b. | It is the only method that will pick up all whitespace characters |
| c. | It is the only one to return a string |
| d. | A and B only |
| e. | A, B, and C |