Started on	Tuesday, 12 April 2022, 2:54 PM
State	Finished
Completed on	Tuesday, 12 April 2022, 3:13 PM
Time taken	19 mins 44 secs
Marks	12.00/15.00
Grade	**80.00** out of 100.00

```
abstract class demo
{
    public int a;
    demo()
    {
        a = 10;
    }

    abstract public void set();

    abstract final public void get();

}

class Test extends demo
{

    public void set(int a)
    {
        this.a = a;
    }

    final public void get()
    {
        System.out.println("a = " + a);
    }

    public static void main(String[] args)
    {
        Test obj = new Test();
        obj.set(20);
        obj.get();
    }
}
```

Select one:

a.

```
None of the above
```

[✓ b.]

```
Compilation error
```

c.

```
a = 20
```

d.

```
a = 10
```

```
The most appropriate matching for the following pairs

X: m=malloc(5); m= NULL;        1: using dangling pointers
Y: free(n); n->value=5;              2: using uninitialized pointers
Z: char *p; *p = 'a';                   3. lost memory is:
```

Select one:

a.

```
X—1 Y—3 Z-2
```

b.

```
X—2 Y—1 Z-3
```

[✓ **c.**]

```
 X—3 Y—1 Z-2
```

d.

```
X—3 Y—2 Z-1
```

```
#include <stdio.h>
int main()
{
    static int i=5;
    if(--i){
        main();
        printf("%d ",i);
    }
}
```

Select one:

a.

```
Compiler Error
```

b. 1 2 3 4

[✓ **c. 0 0 0 0**]

d. 4 3 2 1

```
#include <stdio.h>
struct sample {
    int a = 0;
    char b = 'A';
    float c = 10.5;
};
int main()
{
    struct sample s;
    printf("%d, %c, %f", s.a, s.b, s.c);
    return 0;
}
```

Select one:

○ **a.**

```
No Error, No Output
```

○ **b.**

```
0, A, 10.500000
```

○ **c.**

```
0, A, 10.5
```

[✓ **d. error**]

```
#include <stdio.h>
int fun(char *str1)
{
  char *str2 = str1;
  while(*++str1);
  return (str1-str2);
}

int main()
{
  char *str = "abcdefghi";
  printf("%d", fun(str));
  return 0;
}
```

Select one:

- **a. 10**
- **b. 8**
- [✓ **c. 9**]
- **d.**

```
Random Number
```

```
You are given a list of 5 integers and these integers are in the range from 1 to 6.
There are no duplicates in list.
One of the integers is missing in the list.

Which of the following expression would give the missing number.
^ is bitwise XOR operator.
~ is bitwise NOT operator.

Let elements of list can be accessed as list[0], list[1], list[2], list[3], list[4]
```

Select one:

○ **a.**

```
list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4] ^ 1 ^ 2 ^ 3 ^ 4 ^ 5
```

○ **b.**

```
list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4] ^ 1 ^ 2 ^ 3 ^ 4 ^ 5 ^ 6
```

[✓ **c.**]

```
list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4]
```

○ **d.**

```
~(list[0] ^ list[1] ^ list[2] ^ list[3] ^ list[4])
```

```
What change should be made to code work

struct node
{
int data;
struct node *next;
};

struct node *head=0;

void create(int ele)
{
struct node *nptr,*tptr;
nptr=(struct node *)malloc(sizeof(struct node));
nptr->data=ele;
if(head==0)      // Line A
 tptr=head=nptr;
else
tptr->next=nptr;
tptr=nptr;
nptr->next=0;   // Line B
}
```

Select one:

- **a.** Compiles fine but resutls in segementation fault
- **b.** If we replace Line A and Line B from 0 to NULL the code works
- **c.** Compilation fails

d.

```
No Changes required the code works fine
```

```
Consider the following two sequences :

X = < B, C, D, C, A, B, C >, and
Y = < C, A, D, B, C, B >

The length of longest common subsequence of X and Y is :
```

Select one:

a. 4

b. 3

[✓ **c. 2**]

d. 5

```
Select the appropriate code which tests for a palindrome.
```

Select one:

a.

```
public static void main(String[] args)
{
        System.out.print("Enter any string:");
        Scanner in=new Scanner(System.in);
        String input = in.nextLine();
        Stack<Character> stk = new Stack<Character>();
        for (int i = 0; i < input.length(); i++)
        {
            stk.push(input.charAt(i));
        }
        String reverse = "";
        while (!stk.isEmpty())
        {
            reverse = reverse + stk.pop();
                        stk.pop();
        }
        if (input.equals(reverse))
        System.out.println("palindrome");
        else
            System.out.println("not a palindrome");
}
```

○ **b.**

```
public static void main(String[] args)
{
        System.out.print("Enter any string:");
        Scanner in=new Scanner(System.in);
        String input = in.nextLine();
        Stack<Character> stk = new Stack<Character>();
        for (int i = 0; i < input.length(); i++)
        {
            stk.push(input.charAt(i));
        }
        String reverse = "";
        while (!stk.isEmpty())
        {
            reverse = reverse + stk.pop();
                        stk.pop();
        }
        if (!input.equals(reverse))
        System.out.println("palindrome");
        else
            System.out.println("not a palindrome");
}
```

○ **c.**

```
public static void main(String[] args)
{
        System.out.print("Enter any string:");
        Scanner in=new Scanner(System.in);
        String input = in.nextLine();
        Stack<Character> stk = new Stack<Character>();
        for (int i = 0; i < input.length(); i++)
        {
            stk.push(input.charAt(i));
        }
        String reverse = "";
        while (!stk.isEmpty())
        {
            reverse = reverse + stk.pop();
        }
        if (input.equals(reverse))
        System.out.println("palindrome");
        else
        System.out.println("not a palindrome");
}
```

◯ **d.**

```
public static void main(String[] args)
{
        System.out.print("Enter any string:");
        Scanner in=new Scanner(System.in);
        String input = in.nextLine();
        Stack<Character> stk = new Stack<Character>();
        for (int i = 0; i < input.length(); i++)
        {
            stk.push(input.charAt(i));
        }
        String reverse = "";
        while (!stk.isEmpty())
        {
            reverse = reverse + stk.peek();
        }
        if (input.equals(reverse))
        System.out.println("palindrome");
        else
            System.out.println("not a palindrome");
}
```

```
class Test {
    public static void main(String[] args) {
      for(int i = 0; 0; i++)
      {
          System.out.println("Hello");
          break;
      }
    }
}
```

Select one:

○ **a.**

```
Empty Output
```

b. hello

c.

```
Runtime error
```

[✓ d.]

```
Compiler error
```

```
import static java.lang.System.*;

class StaticImportDemo
{
   public static void main(String args[])
   {
        out.println("welcome to programming");
   }
}
```

Select one:

a.

```
 None of the above
```

[✓ b.]

```
welcome to programming
```

c.

```
Compiler Error
```

d.

```
 Runtime Error
```

```
#include <stdio.h>
int var = 20;
int main()
{
    int var = 5;
    int value = var;
    printf("%d ", value);
    return 0;
}
```

Select one:

[✓ **a. 5**]

b.

```
Garbage Value
```

c.

```
Compiler Error
```

d. 20

```
Consider the polynomial p(x) = a0 + a1x + a2x^2 +a3x^3, where ai != 0, for all i.
The minimum number of multiplications needed to evaluate p on an input x is:
```

Select one:

- a. 4
- [✓ b. 3]
- c. 9
- d. 6

```
class Test
{
        public static void main (String[] args)
        {
                char arr1[] = {'1'};
                char arr2[] = {'1'};
                if (arr1 == arr2)
                        System.out.println("Same");
                else
                        System.out.println("Not same");
        }
}
```

Select one:

- a. same
- [✓ b. not same]
- c.

```
None of these
```

○ **d.**

```
Compilation fails
```

Following is C like pseudo code of a function that takes a number as an argument, and uses a stack S to do processing.

```
void fun(int n)
{
    Stack S;  // Say it creates an empty stack S
    while (n > 0)
    {
      // This line pushes the value of n%2 to stack S
      push(&S, n%2);

      n = n/2;
    }

    // Run while Stack S is not empty
    while (!isEmpty(&S))
      printf("%d ", pop(&S)); // pop an element from S and print it
}
```

What does the above function do in general?

Select one:

○ **a.**

```
Prints binary representation of n in reverse order
```

○ **b.**

```
No Error, No Output
```

```
Prints the value of Logn in reverse order
```

[✓ c.]

```
 Prints binary representation of n
```

d.

```
Prints the value of Logn
```