BITS-Pilani, K.K.Birla Goa Campus

Midsem Exam, 2022-2023 , Object Oriented Programming (CS F213)

Date: 14/03/23 Max. Marks: 50 (30%), Closed Book Time: 9:00 am. to 10:30 am. [Total pages:6]

Instructions: (1) Calculator is not allowed. (2) Answer all questions serially and all parts of the same question together. (3) Make an index page. (4) In questions, where output is asked, first check the program for any errors. If error, state as such and explain the error. If no error, give the output exactly as it will appear on the console (5) All questions have to be answered as per JDK 15.0.1 version, Windows OS. (6) Overwritten answers will not be evaluated. Answer in a legible handwriting.

Q1. Read the program given below and answer the questions that follow: **[6]**

```
1  class Show
2  {
3    static void display(String[] greetings)
4    {
5      for (int i = 0; i <= args.length; i++)
6      {
7        System.out.print(greetings[i] + " ");
8        return true;
9      }
10   }
11 }
12 public class Q1
13 {
14   public static void main(String[] args)
15   {
16     display(args);
17   }
18 }
```

// If I remove extra line out of Bounds exception run time error.

// Show.display(args)

(i) What do you name the file (What do you save the file as) ? Q1

(ii) Give the complete command that you type to compile from the command prompt.

(iii) What is the name of the file that gets created when you compile the code?

(iv) The above code will not compile as written. There are compile time errors. Point out the line nos. at which the errors occur and correct them.

(v) Assume that you corrected the errors and compiled the program. Now that it is compiled, what command do you type to run the program with the following arguments? I LOVE OOP

(vi) Give the exact output that will appear after (v) above

(vii) At which stage is bytecode generated, after compilation or execution? Is it a machine code? — after compilation it is generated

source code → bytecode → machine code

Q2. Consider a class Car which has the following private attributes of the types indicated in brackets: Mileage (float), Model (String). It does not have any method. Consider you already have an object, car1 of class Car in the main class 'A'. Now you have to make another object, car2 of class Car, with exactly the same values as that of car1. But remember that they are still 2 different objects, i.e., any change in one should not affect the other. Write a code snippet to implement this. Clearly indicate the respective classes in which the snippets have to be included. **[3]**

Q3. Read the code given below and answer the questions that follow: **[6]**

```
1  class BITS
2  {
3    boolean x;
4    private int w;
5    static byte student=100;
6    protected char c = 'c';
7  }
8
9  class GOA extends BITS
10 {
11 }
12
13 public class Q3
14 {
15   public static void main(String[]args){
16   }
17 }
```

Write an appropriate nested class EEE (inside class BITS). Write a method static void show() in EEE which performs the following tasks: (Note: Write down only the required code snippets and not the entire code. You cannot declare any additional variables in your code, except object).

i) Increase student by 1 and display the correct result. (Note: You cannot use compound assignment +=, ++prefix or postfix++ operators)

ii) Display the ASCII value of the character stored in c.

iii) Write the line(s) of code to invoke show() from main.

iv) From class EEE, can you access w ? If yes, how? Give the complete statemen? If no, why not?

v) What is this representation '\u0061' called ? What is the base of the number system that it uses.

unicode UTF-16

Q4. . i) Give the output for the code given below. Explain your answer in one line [2]

```
1  class Car
2  {
3      String s;
4      void display()
5      {
6          System.out.println("Hi");
7      }
8  }
9
10 class Honda extends Car
11 {
12     void carcolor(String s)
13     {
14         this.s=s;
15         System.out.println(s);
16     }
17 }
18
19 public class Q4
20 {
21     public static void main(String args[])
22     {
23         Car car1 = new Car();
24         Car City = new Honda();
25         City.carcolor("Platinum White Pearl");
26     }
27 }
```

undefined for type Car

use ((Honda) City).carcolor("Platinum White Pearl"); virtual + desired virtual base

ii) Abstract classes provide 100% abstraction. (True/False) ? [0.5]

iii) There's no way that Abstract classes can be instantiated (True/False)? [0.5]

Q5 i) Give the value of b when the following is executed and displayed: [1]

```
int i = -129;
byte b = (byte) i;
```

-130 -129 [-128 0 127] 128 129

126 127 -128 -127

ii) Consider the expression: c = a+ b; where a=2000000000, b=1000000000. All attributes (a,b,c) are of type int. Will the expression give correct output? If Yes, give the output. If No, modify it to give the correct output. [1]

iii) Is it possible to create an array with mixed type of primitives as given below? If yes, write a code snippet for the following elements. If no, why not? [1.5]

arr[0] = "myname"
arr[1] = 10
arr[2] =15.5

Object[] x = new Object[] {1, 2, 3, "srk"};

Object[] arr = new Object[6]
arr[0] = new String("FP")
arr[1] = new Integer(1);
arr[2] = new String("SP")
arr[3] = new Integer(2);

iv) What is the command 'System.gc ( )' used for? [1]

runs garbage collector expend effort towards recycling unused objects in order to make memory available for quick reuse

v) If a class does not have a superclass, it has by default the ...no-argument constructor... superclass [1]

Q6. Give and explain the output. [1]

```
1  class Birds
2  {
3      String sound(String ...vargs)
4      {
5          System.out.println("Hey this is Robin from Birds class");
6      }
7
8      String sound(String s, String ...vargs)
9      {
10         System.out.println("Hey this is Sparrow from Birds class");
11     }
12 }
13 class Q6
14 {
15     public static void main( String[] args )
16     {
17         Birds bird = new Birds();
18         bird.sound("chuckle", "chirp");
19     }
20 }
```

compiler is aware of what method we are calling

Q7 i) In Overridden methods, return type may or may not be same (True/False) True

ii) With Overloading, the method to be called is determined at compile-time or run time? Give reason for your answer.

iii) Read the program given below. (a) Give the output. (b) Now comment line no.3 and give the output. [3.5]

```
1  class Mechanic {
2      static int parts_sum;
3      static
4      {
5          for(int i=0;i<3;i++)
6              parts_sum+=i;
7          System.out.println(parts_sum);
8      }
9  }
10
11 public class Q7{
12     public static void main(String[] args)
13     {
14         System.out.println (Mechanic.parts_sum);
15     }
16 }
```

Static blocks in Java are executed automatically when the class is loaded in memory. are executed first.

(iii) (a) 0 1 3 / 3 (b) 0

[1]

Q8. Give and explain the output.

```
1  class OOP {
2      public void add(double...args)
3      {
4          int sum = 0;
5          for(int x: args)
6          {
7              sum += x;
8          }
9          System.out.println(sum);;
10     }
11 }
12
13 class Q8{
14     public static void main( String[] args )
15     {
16         OOP OOP1 = new OOP();
17         OOP1.add(2);
18     }
19 }
```

for (double x : args)

Throws error double → int ✗

Q9. (i) Give and explain the output (ii) Now comment line. No.28 and uncomment line.No.29. and explain the output. [2]

```
1  class A
2  {
3    A(int x, double y)
4    {
5      System.out.println("Hey bro, whats'up?");
6      System.out.println(x+" "+y);
7    }
8  }
9  class B extends A
10 {
11   B()   B(int v, double w)
12   {       super(v,w);
13     System.out.println("OOP is interesting");
14   }
15 }
16 class C extends B
17 {
18   C(int a, int b)
19   {
20     super(a,b);
21     System.out.println("I am fine, what about you?");
22   }
23 }
24 class Q9
25 {
26   public static void main(String args[])
27   {
28   //C c = new C(10,12);
29     A a = new A(11,13);
30   }  B b = new B(25, 26);
31 }
```

28

Hey bro, what's up
10 12.0
OOP is interesting
I am fine, what about y

29

Hey bro, what's up
11 13.0

Q10. From line nos. 28 to 35, give the output that will be generated by each of the lines, in addition to the previous line output. If no output, leave a blank for that line no. Write the line nos. and output. [5.5]

```
1  class A
2  {
3    static int x = 2;
4    A()
5    {
6      System.out.println(x);
7    }
8  }
9  class B extends A
10 {
11   int x=3;
12   B()
13   {
14     System.out.println(x);
15   }
16 }
17 class C extends B
18 {
19   C()
20   {
21     System.out.println(x);
22   }
23 }
24 class Q10
25 {
26   public static void main(String args[])
27   {
28     B b = new B();  2 3
29     A a = new A();  2
30     C c = new C();  2,3,3
31     A.x=4;
32     b.x=5;
33     System.out.println(A.x);
34     System.out.println(b.x);
35     System.out.println(c.x);
36   }
37 }
```

30

Hey bro, what's up
25 26.0
OOP is interesting

2 } B.
3
2 } B A
2
3 } C
3
4 } A
5 } B
3 } C

Q11. From line nos. 34 to 40, give the output that will be generated by each of the line, in addition to the previous line output. If no output, leave a blank for that line no. Write the line nos. and output. **[4]**

```
1   class A {
2
3       public static void OOP (String s)
4       {
5           System.out.println("Java");
6       }
7   }
8   class B {
9       String s= "OOP";
10      char c='A';
11      public void OOP (String s)
12      {
13          this.s=s;
14          System.out.println(s);
15      }
16  }
17  class C extends A {
18      public static void OOP (String x)
19      {
20          System.out.println(x);
21      }
22  }
23  class D extends B {
24      char c = 'B';
25      public void OOP (String x)
26      {
27          System.out.println("213");
28          System.out.println(s);
29      }
30  }
31
32  class Q11 {
33  public static void main(String[] args)
34      A a = new C();
35      a.OOP("CS");
36      B b = new B();
37      b.OOP("Hi");
38      B b1 = new D();
39      b1.OOP("F");
40      System.out.println(b1.c);
41      }
42  }
```

Java
Hi
213
OOP
A

(i) Inheritance connection
(ii) All abstract methods in the parent class should be overridden in child class
(iii) If the declared methods are static or final they cannot be overridden

Q12. Give the output. (Code on RHS.) **[2]**

```
class Myclass
{
    int value;
    Myclass(int v)
    {
        value = v;
    }

    static void F1(Myclass u, int v)
    {
        u.value = 12;
        v = 12;
    }
}
class Q12
{
    public static void main(String[] args) {
    Myclass s = new Myclass(20);
    int t = 13;
    Myclass.F1(s,t);;
    System.out.println(s.value + " " + t);
    }
}
```

12 13

Q13 (i) Read the following code and give the output.

```
1  class A {
2    int val;
3
4      A(int _val)
5      {
6        val = _val;
7        System.out.println(val);
8      }
9    void output()
10   {
11     System.out.println(val);
12   }
13 }
14 class B extends A {
15   int val;
16   B(int _val)
17   {
18     super(_val);
19     val = _val;
20   }
21   B(int v1, int v2)
22   {
23     super(v1 + v2);
24   }
25   void output()
26   {
27     System.out.println(val);
28     super.output();
29   }
30 }
31 public class Q13 {
32   public static void main(String[] args) {
33     A p0 = new A(3);
34     A p1 = new B(5);
35     A p2 = new B(6, 7);
36     B p3 = new B(8, 9);
37 // Code snippet goes HERE
38
39   }
40 }
```

		Base	derived
member	b.ch	✓	✗
function	b.f()	✗	✓

3 ✓ 5 ✓ 13 ✓ 17 ✓

3	5 / 5	✓
✗	13 / 17	✗

3

5

13

17

(i) 3 | (ii) 5, 5

(iii) ✓ (iv) Error / mismatch

(v) 0 (not 17)

17

(ii) For each of the following snippets, assume that the snippet is inserted one at a time in the place indicated into the main method. Give the output/error that it produces. If error, explain. Assume that each snippet is tested independently of the others. In other words when you insert the next snippet, comment the previous one. (Note: in case your output contains part repeated from previous bit, then write 'same as bit no.', followed by the additional output that is being generated by the insertion of the snippet. **[6.5]**

a) p0.output();
b) p1.output();
c) p2=p3;
d) p3=p2;
e) p3.output();

Q14. Give the output . **[1]**

[code on RHS] →

```
1  class A {
2      int m;
3      A(int m1)
4      {
5        m = m1;
6      }
7  }
8  public class Q14 {
9      public static void main (String args[])
10     {
11       final A a1 = new A (5);
12       a1.m= 6;
13       System.out.println(a1.m);
14     }
15 }
```

6.

NOT.

final causes ref to change.

data members can be changed too.