

Seat No.: _____

Enrolment No. _____

GUJARAT TECHNOLOGICAL UNIVERSITY
MCA SEM-III Examination- Dec.-2011

Subject code: 630002

Date: 14/12/2011

Subject Name: Fundamentals of Java Programming (Java)

Time: 02.30 pm-05.00 pm

Total marks: 70

Instructions:

1. Attempt all questions.
2. Make suitable assumptions wherever necessary.
3. Figures to the right indicate full marks.

- Q.1 (a)** State following are true or false with justification. **07**
1. A single Java file can contain one or more Java applications.
 2. The size of char data type is 1 byte.
 3. The toString() method can be overridden to be protected.
 4. Java is one platform.
 5. We can create our own sub-class of the java.lang.Math class.
 6. The constructors of enum type are always private.
 7. main() method is required to be written in applet.
- (b)** Attempt the followings. **07**
1. Why compilation of following program gives an error.
abstract class TestAbstract
{
 abstract TestAbstract();
}
 2. A method in Interface is always abstract. Why?
 3. When no constructor defined in class then also object of such class can be created. Why?
 4. abstract and final can not be used together. Justify.
 5. Class defined in Java without using extends keyword having parent(super) class. Why?
 6. It is not compulsory to import java.lang package in Java source file. Why?
 7. After execution of following statements, value of b is not 333. Why?
int i = 333;
byte b = (byte)i;
- Q.2 (a)** Discuss in brief. (Approximately 50 words)
1. Any two features of Java. **02**
 2. Any two primitive data types. **02**
 3. Border Layout manager. **02**
 4. Any two required attributes of Applet tag. **01**
- (b)** Answer the following. (Approximately 50 words for 1,2,3)
1. Differentiate character stream and byte stream. **02**
 2. Define checked and unchecked exception. **02**
 3. Discuss instance of operator in brief. **02**
 4. List any two AWT components. **01**
- OR**
- (b)** Answer the following. (Approximately 50 words for 1,2,3)
1. What is use of import statement? **02**

2. Differentiate throw and throws. **02**
3. Discuss Piped streams in brief. **02**
4. What is meta-annotation? **01**
- Q.3 (a)** Explain the following. (Approximately 100 words)
1. Applet life cycle. **03**
2. Thread life cycle. **04**
- (b)** Answer the following. (Approximately 50 words for 2,3)
1. Give four differences : catch Vs. finally **02**
2. Discuss use of super keyword. **02**
3. Discuss enum data type in brief. **03**
- OR**
- Q.3 (a)** Write short notes on following. (Approximately 100 words)
1. Random Access File **03**
2. Event delegation model. **04**
- (b)** Answer the following. (Approximately 50 words for 2,3)
1. Give four differences : abstract class Vs. interface **02**
2. Discuss use of synchronized keyword. **02**
3. Discuss collection framework in brief. **03**
- Q.4 (a)** Write program for following.
1. To display “Hello India” on frame. **04**
2. To display weekdays along with its description using enum data type. **03**
- | Weekdays | Description |
|-----------|-------------|
| Sunday | Sun |
| Monday | Moon |
| Tuesday | Mars |
| Wednesday | Mercury |
| Thursday | Jupiter |
| Friday | Venus |
| Saturday | Saturn |
- (b)** Write program for following.
1. To display contents of text file on console if file exist. Provide file name from command-line argument. **04**
2. To find average of values given as a command-line arguments. Also write necessary exception handling code. **03**
- OR**
- Q.4 (a)** Write program for following.
1. To display command-line arguments. If command-line arguments are not provided then generate custom exception. **04**
2. To draw a circle in centre of the applet. Circle remains in centre when applet resized. **03**
- Q.4 (b)** Write program for following.
1. To copy file using character stream. Provide source file name and destination file name using command-line arguments. **04**
2. To create two different threads. One thread to display odd numbers between 1 to 40 at every 1 second and other thread to display even numbers between 1 to 20 at every 2 seconds. **03**
- Q.5 (a)** Write corrected code for following. Give reason for errors. Write output of corrected code.

1. class Test2 04
- ```
{
 public static void main(String args[5])
 {
 try
 {
 System.out.println("Test of try");
 }
 }
}
```
2. class Test1 03
- ```
{
    public static void main(String args[])
    {
        testMethod();
    }
    void testMethod()
    {
        System.out.println("Test Method");
    }
}
```
- (b) Write output for following.
1. class Test5 04
- ```
{
 byte b;
 short s;
 int i;
 long l;
 public static void main(String args[])
 {
 Test5 tobj = new Test5();
 System.out.println("byte = " + tobj.b);
 System.out.println("short = " + tobj.s);
 System.out.println("int = " + tobj.i);
 System.out.println("long = " + tobj.l);
 }
}
```

2. class Test6

03

```
{
 public static void main(String args[])
 {
 int a[] = {1, 2, 3};
 int b[];

 System.out.println("Array a");
 for(int i=0; i<a.length; i++)
 {
 System.out.println(a[i]);
 }
 b=a;
 System.out.println("Array b");
 for(int i=0; i<b.length; i++)
 {
 System.out.println(b[i]);
 }
 System.out.println("Array a");
 for(int i=0; i<a.length; i++)
 {
 System.out.println(a[i]);
 }
 }
}
```

**OR**

**Q.5 (a)** Write corrected code for following. Give reason for errors. Write output of corrected code.

1. interface I1

04

```
{
 int v1=5;
}
interface I2 implements I1
{
 int v2;
}
class Test4 implements I2
{
 public static void main(String args[])
 {
 System.out.println(v1);
 System.out.println(v2);
 }
}
```

2. protected class Test3

03

```
{
 public static void main(String args[])
 {
 int i=5;
 String s1="Test3";
 System.out.println(s1+i);
 }
}
```

(b) Write output for following.

1. class RoundTest

04

```
{
 public static void main(String args[])
 {
 System.out.println(Math.round(9.01));
 System.out.println(Math.round(9.5));
 System.out.println(Math.round(-9.5));
 System.out.println(Math.round(-0.1));
 }
}
```

2. class TestP

03

```
{
 TestP()
 {
 System.out.println("Test P");
 }
}
class TestC extends TestP
{
 TestC()
 {
 super();
 System.out.println("Test C");
 }
}
class TestProg
{
 public static void main(String args[])
 {
 TestC tobj = new TestC();
 }
}
```

\*\*\*\*\*