ITS1010 - Programming Fundamentals – Assignment 03
Chathurya Buddhini

1.primitive data types - predefined by the language and is named by a reserved
keyword.

types	sizes	data ranges
byte	8bits	-128 to 127
short	16bits	-32768 to 32767
int	32bits	-2147483648 to 2147483648
long	64bits	-9223372036854775808 to-9223372036854775807
float	32bits	+/-1.4E-45 to +/-3.4028235E+38,+/-infinity,+/-0,NaN
double	64bits	+/-4.9E-324 to +/-1.7976931348623157E+308,+/-infinity,+/-,NaN
char	16bits	\u0000 to\uFFFF
boolean	1 bit used in 32 bit integer	no

----
2. A. byte b1=100;   data ranges is -128 to 127
   C. byte b3=-128;   data ranges is -128 to 127
   D. byte b4=0;   data ranges is -128 to 127
   E. short s1=100;   data ranges is -32768 to 32767
   G. short s3=32767;   data ranges is -32768 to 32767

3.A .char c1='A'; char can store a character
   B. char c2='7'; char can store a character

   D. boolean b1=true;   for this ,java has a boolean data type,which can only
take the values true or false

   E. boolean b2=False;   for this ,java has a boolean data type,which can only
take the values true or false

   F. boolean b3=false;   for this ,java has a boolean data type,which can only
take the values true or false

   G. boolean b4=True;   for this ,java has a boolean data type,which can only
take the values true or false

   H. boolean b5="false";   for this ,java has a boolean data type,which can only
take the values true or false

4.

	binary	octal	hexadecimal
A. 10	1010_2	12_8	A_{16}
B. 16	10000_2	20_8	10_{16}
C. 128	10000000_2	400_8	100_{16}
D. 255	11111111_2	377_8	FF_{16}
E. 32767	111111111111111_2	37777_8	$3FFF_{16}$
F. 1	1_2	1_8	1_{16}
G. 0	0_2	0_8	0_{16}
H. 26	11010_2	32_8	$1A_{16}$
I. 31	10000_2	37_8	10_{16}

5.
A. -10	11110110_2	
B. -100	10011100_2	
C. -64	11000000_2	
D. -1	11111111	
E. -2	11111110_2	
F. -128	10000000_2	
G. 0	0000000	
H. -127	01111111_2	
I. -32	11100000_2	

----

6. a. Conversion and Casting –
   *Conversion
   In type conversion, the source data type with a smaller size is converted into the destination data type with a larger size.

```
Ex class Main {
    public static void main(String[] args) {
    int intType = 20;
     // float is of higher data type than int
    float floatType = intType;

  System.out.println("intType: "+intType);
  System.out.println("floatType: "+floatType);
 }
}
```
Output:
intType: 20
floatType: 20.0

   *Casting
   In type casting, the source data type with a larger size is converted into the destination data type with a smaller size.

```
Ex class Main {
    public static void main(String[] args) {
      intType = 20;
   // short is of lower data type than int
      short shortType = (short)intType;
  System.out.println("intType: "+intType);
  System.out.println("shortType: "+shortType);
 }
}
```
Output:

intType: 20
shortType: 20

b.Narrow Conversion and Narrow Casting
        Narrow conversion
*Automatically putting a big value into a small data type is called narrow
conversion.

        Narrow casting
*Manually puting a large value into a small daya type is called narrow casting.
c. Wider Conversion and Wider Casting
        Wider conversion
*Automatically putting a small value into a large data type is called wider
conversion.

        Wider casting
*Manually putting a small data type into a large data type is called wider casting.

7.B. char ch='A';
     double d=ch;

  D. double d='A';
     char ch=b;
----
8. class Example{
public static void main(String args[]){
byte b1=10,b2=20,b3;
b3=b1+b2; //Line 1
b3=b1+1; //Line 2
b3=b1*2; //Line 3
short s1=10,s2=20,s3;
s3=s1+s2; //Line 4
s3=s1+1; //Line 5
```

```
s3=s*1; //Line 6
int x1=10,x2=20,x3;
x3=x1+x2; //Line 7
x3=b1+b2; //Line 8
x3=b1+1; //Line 9
x3=b1*2; //Line 10
x3=s1+s2; //Line 11
x3=s1+1; //Line 12
x3=s1*1;
}
}
OUTPUT error
```

---

9. A.   l = 2147483647;
   C.   l = 0xabcd;
   E.   l = 0101010110L;

---

10. E. None of above

---

11. B. 4.0 4.5 4.5 5.0

12.A. char a = '\u0061';
    B. char \u0061 = 'a';

13.A. int a = (int)888.8;
    B. byte x = (byte)1000L;
    C. long l = (byte)100;
    D. byte z = (byte)100L;

14.D. 0 to 216

---

15. A. x=Short.MAX_VALUE;
     C. x=-1;

16.A. 3
   B. -3
   C. 3
   D. -3
   E. 3
   F. -3
   G. 3
----

17. A. Line 1
    B. Line 2
    D. Line 4
----

18. A. The result of the expression (1 + 2 + "3") would be
        the string "33".   true

     B. The result of the expression ("1" + 2 + 3) would be
        the string "15".   false

     C. The result of the expression (4 + 1.0f) would be
        the float value 5.0f.   True

     D. The result of the expression (10/9) would be the
         int value 1.   true

     E. The result of the expression ('a' + 1)
        would be the char value 'b'.   true

19.A. int a = (int)888.8;
   B. byte x = (byte)1000L;
   C. long l = (byte)100;
   D. byte z = (byte)100L;
----

20. A. 17
    B. -10
    C. -17
    D. -3
    E. 7
    F. -3

21.-100
   100
   -100
   -200
   400
   0

22.100101
   104
   104

23.101   100
   102   101
   103   102

24.101   101
   102   102
   103   103

25.100
   100
   100
   101
   102
   103

26.3
   0
   10
   0.0
   1.0999999999999996

27.A. x= a + b;   30
   B. x= a +- b;   -10
   C. x= ++a + b;   31
   D. x= a + b++;   31
   E. x= ++a + b++;   33
   F. x= a++ + b++;   34
   G. x= ++a + ++ b; 38
   H. x= a++ + ++b;   39

28.12-4*2:4

   (12-4)*2:16

   12-(4*2):4

29.A. x = 7%10/2*2;

     Explanation :-First   7%10 = 7

     (When using modular,   if we divide a small number into a big number the answer is small number. It's not divide)

        next   7/2 = 3

        last   3*2 = 6, so the answer is 6.

  B. x= 7 % (10 / 2) * 2;

    Explanation :- According to the BODMAS rule first do the brackets.

    (10/2) = 5

     next solve the modular

     7%5 = 2

     last 2*2 = 4, so the answer is 4.

  C. x= 7 % 10 / (2 * 2);

     Explanation :-According to the BODMAS rule first do the brackets.

       (2*2) = 4

      next solve the modular

         7%10 = 7

      last 7/4 = 1 , so the answer is 1.

  D. x= 7 % (10 / (2 * 2));

    Explanation :-According to the BODMAS rule first do the brackets.

     (2*2) = 4

       next slow the second brackets.

         (10 / 4) = 2

      last 7% 2 = 1, so the answer is 1.

   E. x= 7 % ((10 / 2) * 2);

     Explanation :-According to the BODMAS rule first do the brackets.

        (10/2) = 5

      next slow the second brackets.

        (5*2) = 10

      last 7% 10 = 7, so the answer is 7.

----

30. a.106

   b. 12

   c. 11

   d. 44

31.A. x= a++ + a;   21
   B. x= a + a++;   22
   C. x= ++a + a;   26
   D. x= a + ++a;   27
   E. x= ++a + ++a;   31
   F. x= a++ + a++;   33
   G. x= ++a + a++;   38
   H. x= a++ + ++a;   42

32.303
   306

   404   102