Graduate Diploma in Software Engineering
ITS1010 - Programming Fundamentals – Assignment 04

01) byte: this data type represents a signed 8-bit integer value.It has a range of -128 to 127.

short: this data type represents a signed 16-bit integer value.It has a range of -32768 to 32767.

int: this data type represents a signed 32-bit integer value.It has a range of -2^31 to -2^31-1

long: this data type represents a signed 64-bit integer value.It has a range of -2^63 to 2^63-1.

float: this data type represents a signed 64-bit integer value.it has a range of 1.4*10^-45 to 3.4 *10^38.

double: this data type represents a 64-bit floating point value.It has a range of 4.9*10^-324 to 1.8*10^308.

char: This data type represents a single Unicode character. It size is a 16-bit.It has range of 0 to 65535.

boolean: This data type represents a single bit of information and can only have two values: true or false.

02)

A) **byte b1=100** - This statement is legal.because the alue assigned to the byte variable (b1) within the range of the byte data type(-128 to 127)

B) **byte b2=128 -** this statement is illegal.because as the maximum value that can be assigned to the byte data type is 127.therefore this statement will show as compilation error.

C) **byte b3=-128 -** this statement is legal.because the value assigned to the byte variable(b3) eithin the range of the byte data type(128 to 127)

D) **byte b4=0 -** this statement is legal

E) **short s1=100 -** this statement is legai because the value assigned to the short variable s1 is within the range of the byte data type.

F) **short s2=32768** - this statement is illigal because the maximum value that can be assigned to a short data type is 32767.as the result this statement will show compilation error.

G) **short s3=32767 -** this statement is legai because the value assigned to the sort variable(s3) is within the range of the byte data type(-32768 to 32767)

H) **short s4=-32768** - this statement is legal because the value assigned to the short variable s4 is within the range of the short data type (-32768 to 32767).

03)

A) **char c1 = ‘A’; -** This statement is legal.

B) **char c2 = ‘7’;** - this statement is legal.

C) **char c3 = ‘AB’; -** This statement is an illegal.

D) **boolean b1= true; -** This statement is legal.

E) **boolean b2 = False; -** This statement is an illegal.

F) **boolean b3 = false; -** This is legal statement.

G) **boolean b4 = True; -** This is an illegal statement.

H) **boolean b5=”false”; -** This is an illegal statement.

I) **boolean b6 = 0; -** This is an illegal statement.

04)

(A)10

binary-1010

Octal: 12

Hexadecimal: A

(B)16

Binary: 10000

Octal: 20

Hexadecimal: 10

(C)128

Binary: 10000000

Octal:200

Hexadecimal: 80

(D) 255

Binary: 1111111

Octal: 377

Hexadecimal: FF

(E) 32767

Binary: 111111111111111

Octal: 77777

Hexadecimal: 7FFF

(F)1

Binary: 1

Octal: 1

Hexadecimal: 1

(G)0

Binary: 0

Octal: 0

Hexadecimal: 0

(H)26

Binary: 11010

Octal: 32

Hexadecimal: 1A

(I) 31

Binary: 11111

Octal: 37

Hexadecimal: 1F

05)

A) -10 → 11110110
B) -100 → 10011100
C) -64 → 11000000
D) -1 → 11111111
E) -2 → 11111110
F) -128 → 10000000
G) 0 → 00000000
H) -127 → 10000001
I) -32 → 11100000

06)

conversion and casting

conversion	**casting**
conversion a process of converting a value from one data type to another data type	casting is a specific type of conversion where a value is explicitly converted from one data type to another.

Ex:
class Main {
public static void main(String args[]){
byte b = 10; // 8bit
int x; // 32bit
x = b; //conversion
System.out.println(x + " "+ b);
}
}
	Ex:
	class Main {
	public static void main(String args[]){
	int x = 10; //32bit
	short s; //16bit
	//s = x; //Illegal
	s = (short)x; //casting -> Legal
	System.out.println(x + " " + s);
	}
	}

narrow conversion and narrow casting

narrow conversion	**narrow casting**
Narrow conversion occurs when a value of a larger data type is converted to a value of a smaller data type,	narrow casting is a specific type of narrow conversion where a value of a larger data type is explicitly converted to a value of a smaller data type.

Ex:
class Main {
public static void main(String args[]){
double d = 10;
int x = 20;
x += d; //narrow conversion -> x = x + d;
System.out.println(x);
}
}
	Ex:
	class Main {
	public static void main(String args[]){
	double d = 10;
	int x;
	x = (int)d; //narrow casting
	System.out.println(x);
	}
	}
	(double=64 bit,int size= 32 bit)

wider conversion and wider casting

wider conversion	**wider casting**
Wider conversion occurs when a value of a smaller data type is converted to a value of a larger data type.	wider casting is a specific type of wider conversion where a value of a smaller data type is explicitly converted to a value of a larger data type

Ex:
class Main {
public static void main(String args[]){
byte b = 10;
int x;
x = b; //wider conversion
}
}
(byte=8 bit , int=32 bit**)**
	Ex:
	class Main {
	public static void main(String args[]){
	int total = 567;
	int count = 10;
	double avg = total / (double)count;
	// wider casting
	System.out.println(avg); //56.7
	}
	}

07)

(A) double d='A';
char ch='A';

(B) long 1=(int)d;
double d=ch;

(F) float f=65;
int x=(char)f;

08)

line 1,line 2,line 3,line 4,line 5,line 6 ara error.
therefore , not compile and run

09)

10)

11)
E) 4.0,4.5,4.5,5.0

12)
A. char a = '\u0061';
B. char \u0061 = 'a';
C. ch\u0061r a = 'a';

13)
A , C,
B,D codes are legal but i can’t five expecting result.

14) A

15) C. X=-1

16)
3
-3
3
-3
3
-3
3

17)line 1

18)
A
C
D

19)
A. This line code is legal.
B. this line of code is legal.but it will result in data loss.
C. this code is legal.but it will result in compilation error.

20)

```
17
-10
-17
-3
7
-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; - value is 30.

b) x= a +- b; - 10+(-20)=-10 .value is -10.

c) x= ++a + b; - The prefix ++ operator increments the value of a before it is used in the expression.value is 31.

d) x= a + b++; -value is 30.

e) x= ++a + b++; - The prefix ++ operator increments the value of a before it is used in the expression, so a becomes 11.
11+20=31. value is 31.

f) x= a++ + b++; - value is 30.

g) x= ++a + ++ b; - Both a and b are incremented using the prefix ++ operator before they are used in the expression. So, a becomes 11 and b becomes 21.
11 +21=32. value is 32.

g) x= a++ + ++b; - 10+21=32

28)

12 - 4 * 2 : 4
(12 - 4) * 2 : 16
12 - (4 * 2) : 4

29)

6
4
1
1
7

30)

a)106
b)12
c)11
d)44

31)
a)21
b)20
c)22
d)11
e)23
f)21
g)22
h)22

32)

303
306

404 102