Question 1

What value is assigned to **k** in the code:
int k = 13 / 2 + 25 % 10;

○ 8

✓ ○ 11

○ 11.5

○ 21

Question 2

Which of the code samples correctly declares and fills a **char** variable?

○ char c = "A";

✓ ○ char c;
c = 'X';

○ string s;
s = "P";

○ All of the choices are correct

Question 3

Which of the data sets would cause **input failure** on the code?

```
int k;
double d;
char c;
cin >> k >> d >> c;
```

○ 5 2.7 Z

○ 4.8 2 P

✓ ○ 6 A 8.3

○ 8 2 6

Question 4

Given an amount of cents, which line computes the number of quarters in this amount?

int amount = 323;
int nquarters;

○ nquarters = amount - 25;

✓ ○ nquarters = amount / 25;

○ nquarters = amount % 25;

○ nquarters = amount * 25;

Question 5

Which data type typically is allocated one byte of storage?

○ int

○ double

○ string

✓ ○ char

Question 6

What happens if the user enters 3.14 when the code executes?

```
int num = 0;
cout << "Enter a number: ";
cin >> num;
```

○ Variable **num** retains its initial 0 value

✓ ○ Variable **num** now stores value 3

○ Variable **num** now stores value 3.14

○ A runtime error occurs due to the invalid entry.

Question 7

What is the output of the code?

```
double d = 22.5;
int k = static_cast<int>(d);
int rem = k % 4;
cout << rem;
```

○ 3

✓ ○ 2

○ 0

○ 5

Question 8

What is the output of the code? A decimal point indicates a double result while no decimal represents an integer result in the choices.

```
double x = 2.0;
int m = 15, n = 4;
cout << x + m * n - 1;
```

✗ ○ 61

➡ ○ 61.0

○ 67

○ 67.0

Question 9

Use the _____ output manipulator to display all floating point numbers with a decimal point versus scientific notation.

✗ ○ showpoint

○ left

➡ ○ fixed

○ setprecision

Question 10

Which output stream operator affects only the next entry?

○ left

○ setprecision

✓ ○ setw

○ All choices affect only the next entry

Question 11

What is the output of the code? Underscore (_) represents a blank.

```
cout << left << setw(4) << 25 << endl;
cout << setw(4) << 35 << endl;
cout << right << setw(4) << 45 << endl;
```

○
```
2 5 _ _
_ _ 3 5
_ _ 4 5
```

✓ ○
```
2 5 _ _
3 5 _ _
_ _ 4 5
```

○
```
_ _ 2 5
_ _ 3 5
_ _ 4 5
```

○
```
2 5 _ _
3 5 _ _
4 5 _ _
```

Question 12

Code asks the user for a full name which may or may not include blanks. Which input technique should be used?

```
string name;
```

✓ ○ getline (cin, name);

○ cin.get (name);

○ cin >> name;

○ cin >> name >> name;

Question 13

What is the output of the code?

```
int n = 40;
if (n < 20)  cout <<  "Moe" << endl;
else if  (n != 30) cout << "Larry" << endl;
else  cout << "Curly" << endl;
```

○ Curly

○ Moe

✓ ○ Larry

○ Larry
Curly

Question 14

What is the output of the code?

```
if  (20 < 10) cout << "A" << endl;
if  (5 >= 5) cout << "B" << endl;
```

○ A

✓ ○ B

○ A
B

○ There is no output

Question 15

What is displayed by the code?

```
   int n = 6;
```

```
if  (n < 4)
    cout << "A" << endl;
else if (n < 8)
  cout << "B" << endl;
else
    cout << "C" << endl;
```

○ A

✓ ○ B

○ C

○ B
C

Question 16

The data validation loop below ensures that the user enters character 'Y' or 'N'. It should loop until one of these two characters is entered. What is the correct loop test?

```
char reply;
cout << "Enter Y or N: " ;
cin >> reply;
while ( _________________________ )
{   cout << reply << " is not a valid entry" << endl;
    cout << "Enter Y or N: " ;
    cin >> reply;
}
```

✗ ○ reply != 'Y' || reply != 'N'

○ reply == 'Y' || reply == 'N'

○ reply == 'Y' && reply == 'N'

⇨ ○ reply != 'Y' && reply != 'N'

Question 17

After the code completes, variable **k** stores _____ and variable **d** stores _____.

```
int k = 4;
k++;
double d = k / 2;
```

○ 4, 2.0

✓ ○ 5, 2.0

○ 4, 2.5

○ 5, 2.5

Question 18

When used as a prefix operator (ex. **++value**), the increment operator returns the original value prior to the addition of 1.

○ True

✓ ○ False

Question 19

The user should enter one of two strings: "X" or "Z". Which loop test should be used make the code repeat until a valid entry?

```
char choice;
do
{     cout << "Enter X or Y: ";
      cin >> choice;
} while (_____);
```

✓ ○ choice != 'X' && choice != 'Y'

○ choice != ('X' || 'Y')

○ choice == 'X' && choice == 'Y'

○ choice != 'X' !! choice != 'Y'

Question 20

Which situation causes input failure?

➡ ○ Code is reading an int into variable **k** and the user types: Q

○ Code is reading a double into variable **d** and the user types: 15

○ Code is reading an int into variable **k** and the user types: 3.14

✗ ○ Code is reading a character into variable **c** and the user types: 5