

מבוא לתכנות

הרצאה 2 – ביטויים לוגיים ותנאים

סמטר 1

סוגי נתונים

- סוגי נתונים מוכרים:

- Integer
- Float
- String
- Double

- כיצד לייצג ערך שיכול להיות "כן" / "לא"?

סוג נתון בוליאני

- בC#, "כן" / "לא" מיוצג באמצעות משתנים בוליאניים. למשתנה בוליאני יכול להיות אחד משני הערכים הבאים:

– אמת

– שקר

```
bool x = true;  
bool y = false;
```

אופרטורים יחסיים

- האופרטורים שראינו עד כה:
 - אופרטורים אריתמטיים: +, -, *, /
 - אופרטורים למחרוזת: +
- בC# ישנם אופרטורים שמחזירים ערכים בוליאניים:

אופרטור	משמעות	דוגמה (x=5)
==	שווה?	x == 5 # True
!=	שונה?	x != 5 # False
>	גדול מ?	x > 3 # True
<	קטן מ?	x < 4 # False
>=	גדול או שווה ל?	x >= 5 # True
<=	קטן או שווה ל?	x <= 9 # True

אופרטורים יחסיים

אופרטור	משמעות	דוגמה (x=5)	
==	שווה?	x == 5	# True
!=	שונה?	x != 5	# False
>	גדול מ?	x > 3	# True
<	קטן מ?	x < 4	# False
>=	גדול או שווה ל?	x >= 5	# True
<=	קטן או שווה ל?	x <= 9	# True

- האופרטורים הללו פועלים על 2 אופרנדים.
- אופרנד, כידוע, יכול להיות ערך קבוע, משתנה, או ביטוי מורכב.
- הביטוי שמתקבל, למשל $x \geq 5$, נקרא "ביטוי בוליאני".

הפעלה מותנית – אם if

- אנו רוצים לכתוב תוכנית שממשת תהליך התחברות ע"י זיהוי משתמש.
- לתוכנית יש קלט אחד: שם משתמש.
- אם שם המשתמש הינו ' bob ', אנו נרצה שהתוכנית תדפיס: 'אמת'; בכל מקרה, היא תדפיס ' Goodbye ' בסוף.

```
string username = Console.ReadLine();  
if (username == "bob")  
{  
    Console.WriteLine("True");  
}  
Console.WriteLine("Goodbye");
```

if(ביטוי בוליאני)
{
 הקוד רץ רק עבור המקרה
 בו הביטוי הבוליאני הוא
 אמת.
}

המשך הקוד בכל מקרה

הפעלה מותנית – אם if

- התוכן שבסוגריים {} נקרא בלוק, במקרה שלנו בלוק ה"if".
- יש לבצע אינדטציה (הזחה) לקוד שבתוך ה {}.
- בלוק ה"if" יבוצע רק אם התנאי הבוליאני שווה אמת.
- הקוד מחוץ לבלוקי ה if יבוצע בכל מקרה.
- קיצור דרך: if → Tab → Tab
- השימוש בסוגריים {} לא הכרחי אם רושמים שורת קוד אחת.

```
if(ביטוי בוליאני)
{
    קוד שרץ רק במקרה
    ובו הביטוי הבוליאני
    הינו אמת
}
המשך הקוד בכל מקרה
```

קינוח תנאים

- מה בנוגע לסיסמה?
- נוכל לשנות את התוכנית כך שתקלוט 2 קלטים:
username, password
- אם שם המשתמש הינו 'bob' והסיסמה היא '1234', אנו נרצה להדפיס אמת; בכל מקרה, נדפיס 'Goodbye' בסוף.

```
string username = Console.ReadLine();  
string password = Console.ReadLine();  
if (username == "bob")  
{  
    if (password == "1234")  
    {  
        Console.WriteLine("True");  
    }  
}  
8 Console.WriteLine("Goodbye");
```


אופרטורים לוגיים

- לעתים, אנו נאלצים לשלב את התוצאות של שניים או יותר תוצאות של ביטויים בוליאניים. לדוגמה:
 - `username == 'bob'`
 - `password == '1234'`
- נוכל לחשוב על לפחות 2 דרכים לשילוב ביטויים כאלו:
 - שני הביטויים הם "אמת" (וגם - `and`)
 - לפחות אחד מהביטויים "אמת" (או - `or`)

אופרטורים לוגיים

- #C בעלת 3 אופרטורים לוגיים:

אופרטור	מטרה	דוגמה
$x \ \&\& \ y$	וגם: אמת אם 2 הביטויים אמת; שקר אחרת.	$x \geq 3 \ \&\& \ x < 9$
$x \ \ y$	או: אמת אם ביטוי אחד לפחות הוא אמת; שקר אחרת.	$x < 3 \ \ x \geq 9$
$!x$	היפוך: אמת אם הביטוי הוא שקר, ושקר אם הוא אמת.	$!(x \leq 5)$

- Examples:

– $!(x == 5)$ # same as $x != 5$

– $!(x \geq 3 \ \&\& \ x < 9)$ # same as $x < 3 \ || \ x \geq 9$

אופרטורים לוגיים

```
if (username == "bob")
{
    if (password == "1234")
    {
        Console.WriteLine("True");
    }
}
Console.WriteLine("Goodbye");
```

- איך נוכל לרשום את הקוד באמצעות הצהרת if אחת?

```
if (username == "bob" && password == "1234")
{
    Console.WriteLine("True");
}
Console.WriteLine("Goodbye");
```

- הקוד הנ"ל יעיל וקריא יותר.

אופרטורים לוגיים

- אם נרצה להדפיס "אמת" עבור שם המשתמש והסיסמה הנכונים ו"שקר" אחרת, נוכל להיעזר בקוד הבא:

```
Console.WriteLine(username == "bob" && password == "1234");
```

הפעלה אחרת – אחרת else

- אנו רוצים לכתוב תוכנית שמציגה הודעות שונות, בתלות תוצאת ההתחברות:

– אם ההתחברות צלחה, נציג "Welcome!"

– ואם לאו, נציג "Login failed"

- בכל מקרה, נדפיס 'Goodbye' בסוף.

```
if (username == "bob" && password == "1234")
{
    Console.WriteLine("Welcome!");
}
else
{
    Console.WriteLine("Login Failed!");
}
Console.WriteLine("Goodbye");
```

if(ביטוי בוליאני)

{

קוד שרץ עבור אמת

else

{

קוד שרץ עבור שקר

}

המשך הקוד בכל מקרה

הפעלה מותנית – else

- התוכן שבסוגריים {} נקרא בלוק, במקרה שלנו בלוק ה"else".
- יש לבצע אינדטציה (הזחה) לקוד שבתוך ה {}.
- בלוק ה"else" יבוצע רק אם התנאי הבוליאני שווה שקר.
- הקוד מחוץ לבלוקי ה if וה else יבוצע בכל מקרה.
- קיצור דרך: if → Tab → Tab
- השימוש בסוגריים {} לא הכרחי אם רושמים שורת קוד אחת.

```
if(ביטוי בוליאני)
{
    קוד שרץ עבור אמת
else
{
    קוד שרץ עבור שקר
}
המשך הקוד בכל מקרה
```

אפשרויות רבות – if else

- ראו מימוש הקוד הבא:

```
if (username == "bob" && password == "1234")
{
    Console.WriteLine("Welcome!");
}
else
{
    Console.WriteLine("Login Failed!");
}
Console.WriteLine("Goodbye");
```

- כעת יש לנו משתמש חדש, טום, אשר סיסמתו: 5678

```
if (username == "bob" && password == "1234")
{
    Console.WriteLine("Welcome bob!");
}
else if (username == "tom" && password == "5678")
{
    Console.WriteLine("Welcome tom!");
}
else
{
    Console.WriteLine("Login Failed!");
}
Console.WriteLine("Goodbye");
```

רק ביטויי האמת הראשון יבוצע,
השאר ידולגו!

מה יקרה כאשר נשתמש רק
בif?

מה יקרה אם נשתמש רק בif
else אבל לא בif else?

אפשרויות רבות

- בואו ניצור תוכנית: בהינתן מספר חודש, נדפיס את שמו.

```
int month_num =  
int.Parse(Console.ReadLine());  
if (month_num == 1)  
    Console.WriteLine("january");  
else if (month_num == 2)  
    Console.WriteLine("February");  
else if (month_num == 3)  
    Console.WriteLine("March");  
else if (month_num == 4)  
    Console.WriteLine("April");  
else if (month_num == 5)  
    Console.WriteLine("May");  
else if (month_num == 6)  
    Console.WriteLine("June");  
else if (month_num == 7)  
    Console.WriteLine("July");
```

מה קורה אם הפרמטר גדול מ12
או קטן מ1?

```
else if (month_num == 8)  
    Console.WriteLine("August");  
else if (month_num == 9)  
    Console.WriteLine("September");  
else if (month_num == 10)  
    Console.WriteLine("October");  
else if (month_num == 11)  
    Console.WriteLine("November");  
else if (month_num == 12)  
    Console.WriteLine("December");
```


אופרטור מודולו - %

- אופרטור % ("מודולו") מחשב את השארית השלמה של חלוקה בין שני מספרים:

$$- 10 \% 2 = 0$$

$$- 15 \% 2 = 1$$

$$- 13 \% 10 = 3$$

- נוכל להיעזר ב% על מנת לבדוק האם מספר מתחלק במספר אחר ללא שארית:

$$- \# \quad 0 == 2 \% \text{אאמת רק אם} \quad X \text{זוגי}$$

$$- \# \quad 0 == 3 \% \text{אאמת רק אם} \quad X \text{מתחלק ב} 3$$

$$- \# \quad 0 == 7 \% \text{אאמת רק אם} \quad X \text{מתחלק ב} 7$$

- כתבו תוכנית: בהינתן מספר n , נדפיס אם n זוגי או אי-זוגי.

– לדוגמה: אם $n = 28$, התוצאה תהיה "28 is Even".

```
int n = int.Parse(Console.ReadLine());  
if (n%2==0)  
{  
    Console.WriteLine(n + " is Even");  
}  
else  
{  
    Console.WriteLine(n + " is Odd");  
}
```

פונקציות מתמטיות

- `double variableName = Math.Sqrt(double variableName)` → שורש $\sqrt{}$
- `double variableName = Math.Pow(double variableName, double variableName)`

→ y בחזקת x

```
double d = 100;  
Console.WriteLine(Math.Sqrt(d));  
//will print 10  
Console.WriteLine(Math.Pow(d/10,3));  
//will print 1000
```

תרגילים

1. כתבו תוכנית אשר קוראת 3 מספרים מהמשתמש ומדפיסה אותם בסדר עולה.

– יש יותר מפתרון אחד...

2. קוראת מהמשתמש 3 ציונים עבור: מתמטיקה, אנגלית ופיזיקה. תחשב ממוצע לפי חלוקה של 40% מתמטיקה, 20% אנגלית ו-40% פיזיקה, ותדפיס אותו.

3. תקבל מהמשתמש אורך ורוחב של החדר שלו ומחזירה את השטח וההיקף שלו.

4. בהינתן משולש ישר זווית, חשבו את היתר על ידי קריאת 2 הצלעות האחרות שלו באמצעות שימוש בנוסחה:

$$a^2 + b^2 = c^2$$