Progoblig 1 (Individuell del)

Før jeg starter… kan jeg spørre om et annet språk??? Please

Ind01

1. I utrykket «3 + 5», er «3» og «5», operander mens «+» er operatoren.
2. Fire eksempler på aritmetiske operatorer er: Multiplikasjon, Divisjon, Addisjon og Subtraksjon.
3. En kompilator kan tolke «3+4\*5» på et par forskjellige måter. Det varierer basert på hvilke kodespråk som blir brukt. I noen språk så vil kompilatoren tolke regnestykke som vi ville gjort det. Altså med PEMDAS. Noen andre språk vil tolke operatorene som like i prioritet og vil derfor trenge at du spesifiserer nøyaktig hva slags rekkefølge du vil ha.
4. CPO så på de to operatorene som like i prioritet eller gruppering, så bruk av PEMDAS måtte bli gjort manuelt ved å skrive parentes på riktig sted fordi regnerekkefølgen er ukjent for CPO
5. JSC tolket mattestykket med PEMDAS og fikk riktig svar med en gang uten at noe måtte bli gjort manuelt.
6. CPO er ukjent med regnerekkefølgen og vil derfor gi en error kode i ulikhet med JSF som vil gi riktig svar fordi JavaScript er et språk som bruker PEMDAS. CPO trenger altså hjelp manuelt for å komme seg frem til riktig svar mens JSF for det meste bare trenger det originale regnestykke.
7. I dette tilfelle må man skrive parentes på riktig sted i begge språkene for å få riktig svar. I dette tilfelle så er situasjonen i CPO det samme som det forrige stykket. Endringen skjer i JSF der man må skrive parentes for å overstyre bruken av PEMDAS der JSF har lik prioritet på subtraksjon og addisjon.
8. Operatørene «+» og «-» blir prioritert på likt nivå og blir regnet ut først ut ifra hvor på linja operatøren forekommer.’
9. Operatørene «\*» og «/» blir prioritert på likt nivå og blir regnet ut først ut ifra hvor på linja operatøren forekommer.
10. A screenshot of a computer

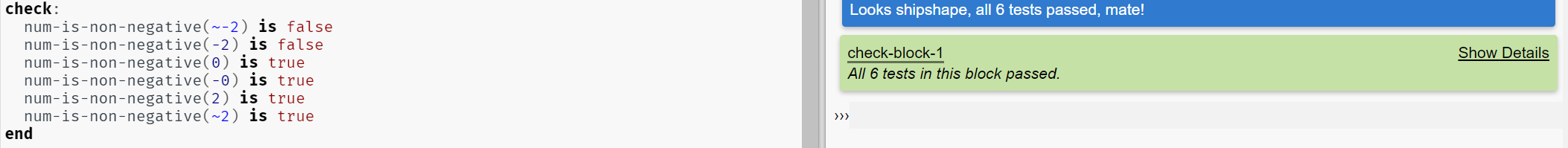
    Description automatically generated

I bildet over blir «4/2» først regnet ut, etterfulgt med «2\*6» før de plusses sammen til 14 der 3 blir trukket fra. Altså slik som bildet under viser.

A screenshot of a math equation

Description automatically generated

1. Ulempene med CPO er at man må manuelt skrive inn rekkefølgen man vil at utrykket skal regnes ut for å få riktig svar. Fordelen er at CPO regner ut nøyaktig hvordan utrykket blir skrevet så du ikke trenger å spesifisere det hvis utrykket du har skrevet er nøyaktig det du lurer på.
2. Hvis man vil bruke andre operasjoner må man skrive den riktige funksjonen og enten «sjekke» om det er riktig for å passere en test eller skrive det utenfor en «sjekk» for å KANSKJE få et svar.



Num-is-non-negative er Pyret sin versjon av «=>»

num-min brukes for å se hvilke av 2 nummeret som er det minste. I likhet brukes num-max for å se hvilke av 2 nummere som er størst.



|  |  |
| --- | --- |
| 3+4 – num-min(7, 4) | utrykk |
| 3+4  3-4  Num-max(-4, 4)  «Hei» | program |
| «IS-114» | verdi |
| 3.141592653 | verdi |
| «Hei på deg» | verdi |
| «håper det går bra med deg»  ‘3’  2 + ’3’ | program |

1. JSF er en konsoll så den godtar bare en linje om gangen. På grunn av dette kan man ikke lage et program i JSF, men JSF klarer fortsatt å holde verdier og å regne ut utrykk som vist under:

A screenshot of a computer

Description automatically generated

1. CPO tolker utrykket som vi ville forventet det. Dette skjer fordi CPO runder av svaret slik at det ikke skjer noen uventede problemer slik som skjer i JSF(Forklarer videre da jeg forklarer JSF). Du kan se et eksempel på dette i bilde under:



JSF gjør det på en annen måte. JSF gir et nøyaktig svar og vil derfor gi svaret 0.30000000000000004. Grunnen til at dette skjer er fordi JSF ikke ser på desimal tall likt som mennesker. For oss gir det mening at 0.33 \* 3 (1/3 \* 3) er 1, men for JSF så vil det bli 0.9999… fordi den ikke for den riktige verdien, den bare nærmer seg. JSF tolker tallene som binær og på grunn av dette så skjer det feil.

A screenshot of a computer code

Description automatically generated

Ind02

1. A white background with black and white clouds

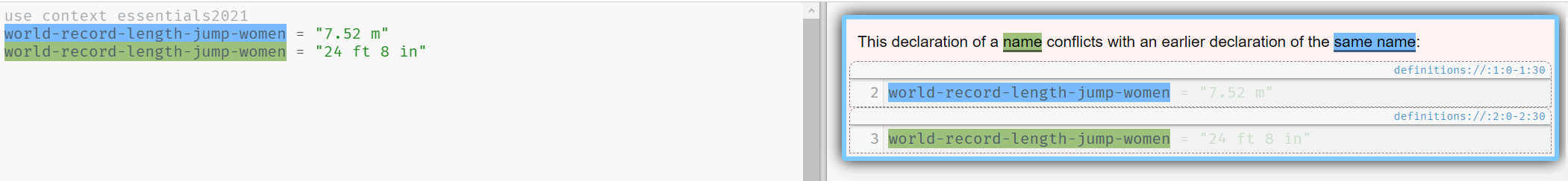
   Description automatically generated with medium confidence
2. Nr.1 er et utrykk fordi det er et stykke som ikke definerer noe.

Nr.2 er en definisjon fordi x blir definert som en verdi.

Nr.3 er et utrykk fordi den får en funksjon til å gjøre noe.

Nr.4 er en definisjon fordi du definerer blue-circ som en sirkel ved hjelp av det forrige utrykket.

1. Da man skriver inn flere variabler med samme navn, får man en error kode. Grunnen til dette er fordi den ikke vet hvilken den skal henvende seg til fordi man ikke kan overstyre en variabel i CPO.



1. Man kan definere variablene annerledes som vist under slik at det ikke skapes en konflikt:

A green text on a white background

Description automatically generated

1. I JSF så overstyrer den nye variabelen den gamle, men JSF liker ikke bindestreker så navnet måtte endres:

A screenshot of a computer

Description automatically generated

1. Definisjonsvinduet er det vinduet man finner på venstre side av CPO. Det er her man skriver inn kode og skriver alle funksjonene man senere vil se i interaksjonsvinduet som man finner på høyre side. s
2. A computer screen shot of a computer code

   Description automatically generated